

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Навчально-науковий центр заочної форми навчання

(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки

(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка системи візуального контролю пошуку дефектів на поверхні

друкованих виробів за технологією FDM/FFF

(тема)

Виконав:

здобувач 2 року навчання,

групи КІПВзм-23-1

Погребняк Вікторія Валеріївна

(прізвище, ініціали)

Спеціальність 174 Автоматизація,

комп'ютерно-інтегровані технології та

робототехніка

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерно-інтегровані

технологічні процеси і виробництва

(повна назва освітньої програми)

Керівник проф. Омаров Ш.А.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри КІТАР

(підпис)

Невлюдов І.Ш.

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Навчально-науковий центр заочної форми навчання

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерно-інтегровані технологічні процеси і виробництва

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

«\_\_\_» \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Погребняк Вікторії Валеріївни  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи візуального контролю пошуку дефектів на поверхні друкованих виробів за технологією FDM/FFF

затверджена наказом університету від 22 листопада 2024 р. № 197Стз

2. Термін подання здобувачем роботи до екзаменаційної комісії 27 січня 2025 р.

3. Вихідні дані до роботи: \_\_\_\_\_

3.1 Методи виготовлення виробів за технологією FFF/FDM 3D-друку;

3.2 Технологія контролю якості 3D-друкованих виробів за допомогою обробки зображень;

3.3 Бібліотека OpenCV для реалізації обробки зображень, Python для розробки основного алгоритму та інтерфейсу програми;

3.4 Дослідження методів обробки зображень (лінеаризація, фільтрація, згладжування) та їх впливу на точність виявлення дефектів у 3D-друкованих деталях.

4. Перелік питань, що потрібно опрацювати в роботі: \_\_\_\_\_

4.1 Вступ;

4.2 Аналіз предметної області;

4.3 Методи обробки зображень для виявлення дефектів у 3D-друці;

4.4 Розробка програмного засобу для пошуку дефектів;

4.5 Охорона праці;

4.6 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій \_\_\_\_\_  
Демонстраційний матеріал представлений у форматі презентації PowerPoint (\*.ppt) – 10 с.  
формату А4 \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

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

#### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання та аналіз завдання	22.11.2024	виконано
2	Збір матеріалів для дослідження	23.11.2024	виконано
3	Розробка розділу «Аналіз предметної області»	30.11.2024	виконано
4	Аналіз літературних джерел з технологій 3D-друку	05.12.2024	виконано
5	Аналіз властивості матеріалів для 3D-друку	10.12.2024	виконано
6	Дослідження налаштування програми-слайсера для роботи з 3D-моделлю та їх вплив на дефекти	18.12.2024	виконано
7	Дослідження впливу різних методів обробки зображень з дефектами	20.12.2024	виконано
8	Розробка програмного засобу для ідентифікації дефектів	22.12.2024	виконано
9	Тестування програмного засобу	02.01.2025	виконано
10	Розробка розділу «Охорона праці»	11.01.2025	виконано
11	Оформлення пояснювальної записки	12.01.2025	виконано

Дата видачі завдання \_\_\_\_\_ 22 листопада 2024 \_\_\_\_\_

Здобувач \_\_\_\_\_

  
(підпис)

Керівник роботи \_\_\_\_\_

(підпис)

проф. Омаров .Ш.А \_\_\_\_\_

(посада, прізвище, ініціали)

Я, як студентка ХНУРЕ Погребняк Вікторія Валеріївна, розумію і підтримую політику закладу із академічної доброчесності. Я не надала і не одержала недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Дата

16.01.25



ПІБ

Погребняк В.В.

## РЕФЕРАТ

Пояснювальна записка: 95 с., 37 рис., 1 таблиця, 22 джерела.

3D-ДРУК, 3D-ПРИНТЕРИ, АВТОМАТИЗАЦІЯ, АЛГОРИТМ, ДЕФЕКТИ, ІНТЕРФЕЙС, КОНТРОЛЬ ЯКОСТІ, ЛІНЕАРИЗАЦІЯ, СЛАЙСЕР, ФІЛЬТРАЦІЯ ЗОБРАЖЕНЬ, FFF/FDM ТЕХНОЛОГІЯ, OPENCV, PYTHON.

Об'єкт дослідження – процес 3D-друку та якість виготовлення тривимірних виробів.

Предмет дослідження – методи автоматизованого аналізу зображень для виявлення дефектів у готових 3D-друкованих деталях.

Мета дослідження – розробка програмного засобу для аналізу зображень 3D-друкованих виробів з метою виявлення дефектів, що дозволить підвищити якість кінцевої продукції та мінімізувати втрати в процесі виробництва.

Проведено аналіз видів технологій 3D-друку та властивості існуючих матеріалів для 3D-друку, досліджено інтерфейс, параметри та можливості програми-слайсера для підготовки 3D-моделі до друку, а також виконано обробку зображень деталей, виготовлених методом 3D-друку для подальшого пошуку дефектів за допомогою програмного засобу.

Наукова новизна даної роботи полягає у розробці інноваційного підходу до автоматизації контролю якості в 3D-друку. У контексті стрімкого зростання популярності 3D-друку, особливо в галузі швидкого прототипування, запропоновано новий підхід до контролю якості. Це особливо актуально для військової сфери, де швидкість впровадження нових рішень є критичною для ефективності.

Запропонований метод контролю якості характеризується простотою, доступністю та економічністю розробленого програмного забезпечення. Традиційно

програмні засоби для контролю якості в промисловості є важкими у використанні, дорогими, вимагають тривалого навчання персоналу та залучення експертів для підтримки. Розроблене програмне забезпечення, засноване на використанні бібліотеки OpenCV, є легким для впровадження та не вимагає високого рівня технічних знань від користувачів.

Також розроблений ПЗ може бути легко інтегровано у виробничі процеси завдяки простому інтерфейсу та можливості роботи з популярними форматами зображень, що дозволяє оптимізувати контроль якості без необхідності масштабної модернізації виробничих ліній.

На сьогодні на ринку відсутні легкі, прості у використанні програмні засоби, які могли б забезпечувати автоматизований контроль якості 3D-друкованих виробів. Запропоноване рішення заповнює цю нішу, забезпечуючи автоматизацію виробничих процесів з мінімальними витратами на впровадження та обслуговування.

Використання результатів даного дослідження дозволяє оптимізувати виробничий процес, а саме:

- швидко виявляти дефекти, зменшуючи кількість бракованих виробів;
- підвищити ефективність процесу прототипування, зокрема у військовій сфері, де час є вирішальним фактором;
- знизити витрати на ручну перевірку та обслуговування обладнання.

Таким чином, робота не тільки демонструє новий підхід до вирішення задачі автоматизації контролю якості у сфері 3D-друку, а й сприяє досягненню Цілі сталого розвитку 9 "Промисловість, інновації та інфраструктура", зокрема завдань 9.2 «Сприяти інклюзивній та стійкій індустріалізації, підвищуючи частку виробництва у ВВП» та завдання 9.4 «Осучаснювати інфраструктуру та технології, що сприятиме підвищенню ефективності виробництва і створенню екологічно чистих виробничих процесів, а також Цілі сталого розвитку 12 "Відповідальне споживання та виробництво", зокрема завдання 12.5 «Зменшити обсяги відходів шляхом запобігання, зменшення, переробки та повторного використання».

## ABSTRACT

Explanatory note: 95 pages, 37 figures, 1 table, 22 references.

3D PRINTING, 3D PRINTERS, ALGORITHM, AUTOMATION, DEFECTS, FFF/FDM TECHNOLOGY, IMAGE FILTRATION, INTERFACE, LINEARIZATION, OPENCV, PYTHON, QUALITY CONTROL, SLICER.

Object of study – the process of 3D printing and the quality of manufactured three-dimensional products.

Subject of study – methods for automated image analysis to detect defects in finished 3D-printed parts.

Research objective – to develop software for analyzing images of 3D-printed products to identify defects, improving product quality and minimizing production losses.

The study includes an analysis of 3D printing technologies and the properties of existing 3D printing materials. It investigates the interface, parameters, and capabilities of slicer software for preparing 3D models for printing. Additionally, it covers the image processing of 3D-printed parts to detect defects using the developed software.

The scientific novelty of this work lies in the development of an innovative approach to automating quality control in 3D printing. Given the rapid growth in the popularity of 3D printing, particularly in rapid prototyping, a new quality control approach is proposed. This is particularly relevant for the military sector, where the speed of implementing new solutions is critical to efficiency.

The proposed quality control method is characterized by the simplicity, accessibility, and cost-effectiveness of the developed software. Traditional quality control software in industry tends to be complex, expensive, require extensive personnel training, and depend on experts for support. The software developed in this research, based on the OpenCV

library, is easy to implement and does not demand a high level of technical expertise from users.

Furthermore, the software can be seamlessly integrated into production processes thanks to its simple interface and compatibility with popular image formats. This optimizes quality control without requiring extensive modernization of production lines.

Currently, the market lacks lightweight, user-friendly software for automated quality control of 3D-printed products. The proposed solution fills this gap by automating production processes with minimal implementation and maintenance costs.

Practical applications of this research enable optimization of the production process:

- quickly identifying defects, reducing the number of defective products;
- increasing the efficiency of prototyping, especially in the military field, where time is critical;
- reducing costs associated with manual inspection and equipment maintenance.

Thus, this work not only demonstrates a novel approach to solving the problem of quality control automation in 3D printing but also contributes to achieving Sustainable Development Goal (SDG) 9: "Industry, Innovation, and Infrastructure," particularly target 9.2 ("Promote inclusive and sustainable industrialization, increasing the share of manufacturing in GDP") and target 9.4 ("Modernize infrastructure and technology to improve efficiency and foster eco-friendly production processes"). It also aligns with SDG 12: "Responsible Consumption and Production," specifically target 12.5 ("Reduce waste generation through prevention, reduction, recycling, and reuse").

## ЗМІСТ

Перелік умовних скорочень .....	12
Вступ.....	13
1 Аналіз предметної області.....	15
1.1 Види технологій 3D-друку .....	15
1.1.1 Типи матеріалів для 3D-друку .....	18
1.1.2 Схема побудови та компоненти 3D-принтера.....	20
1.2 FFF/FDM технологія 3D-друку .....	22
1.2.1 Основні відомості про FFF/FDM технологію .....	22
1.2.3 Принтери, які працюють з технологією FFF/FDM .....	24
1.3 Підготовка матеріалів для друку та налаштування параметрів принтера .....	28
1.3.1 Програми слайсери.....	29
1.3.2 Слайсер Ultimaker Cura.....	31
1.4 Дефекти 3D-друку .....	34
1.4.1 Погана адгезія .....	35
1.4.2 Надмірна, або недостатня екструзія.....	36
1.4.3 Надмірне нагрівання .....	36
1.4.4 Поява павутиння.....	37
1.4.5 Зміщення шарів .....	38
1.4.6 Відсутність подачі пластику .....	38
1.4.7 Розщеплення шарів, утворення тріщин .....	39
1.4.8 Дефект «Слонова нога» .....	40

1.4.9 Дефект «Дзвін» .....	40
1.4.10 Присутність щілин на моделях із тонкими стінками .....	41
1.4.11 Недотримання геометрії .....	42
1.5 Висновки до першого розділу .....	42
2 Методи обробки зображень для виявлення дефектів у 3D-друці .....	44
2.1 Обробка зображень з метою знаходження дефектів .....	45
2.1.1 Лінеаризація зображень .....	45
2.1.2 Фільтрація .....	47
2.1.3 Згладжування .....	49
2.2 Висновки до другого розділу .....	51
3 Розробка програмного засобу .....	53
3.1 Бібліотека OpenCV .....	53
3.2 Аналіз мов програмування, які працюють з бібліотекою OpenCV .....	54
3.2.1 Мова програмування Java .....	55
3.2.2 Мова програмування C++ .....	57
3.2.3 Мова програмування Python .....	58
3.3 Розробка програми на Python та OpenCV .....	59
3.3.1 Розробка основного алгоритму програми .....	60
3.3.2 Розробка інтерфейсу програми .....	61
3.3.3 Подальше розповсюдження ПЗ .....	62
3.4 Тестування програми .....	64
3.4.1 Перевірка позитивного сценарію .....	65
3.4.2 Перевірка негативно сценарію .....	67

3.4.3 Перевірка вибору одного файлу двічі .....	68
3.5 Висновки до третього розділу.....	71
4 Охорона праці.....	73
Висновки .....	75
Перелік джерел посилання .....	77
Додаток А Апробація наукових результатів дослідження.....	80
Додаток Б Код програми для обробки зображень .....	91

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ЦСР – Цілі сталого розвитку;

ЧПУ – Числове програмне управління;

ABS – Acrylonitrile Butadiene Styrene;

CNN – Convolutional neural network;

DLP – Digital Light Processing;

DMLS – Direct Metal Laser Sintering;

EBM – Electron Beam Melting;

FDM – Fused Deposition Modeling;

FFF – Fused Filament Fabrication;

JVM – Java Virtual Machine;

LCD – Liquid-crystal Display;

LDM – Liquid Deposition Modeling;

MATLAB – Matrix Laboratory;

MJF – Multi Jet Fusion;

PETG – Polyethylene terephthalate glycol;

PLA – Polylactic Acid;

SLA – Stereolithography;

SLM – Selective Laser Melting;

SLS – Selective Laser Sintering;

SCARA – Selective Compliance Articulated Robot Arm;

PIL – Python Imaging Library.

## ВСТУП

В сучасному світі технологія 3D-друку швидко набуває популярності і стає невід'ємною частиною багатьох галузей, включаючи медицину, автомобілебудування, авіакосмічну та будівельну промисловість. Вона дозволяє значно скоротити час і витрати на виготовлення складних деталей та зменшує залежність від традиційних виробничих методів. Особливо актуальною ця технологія стає в умовах підвищеної потреби в гнучких, швидких і доступних рішеннях для створення прототипів та кінцевих продуктів. У наш час, коли інновації та автоматизація стають основою економічного зростання, використання 3D-друку надає компаніям стратегічну перевагу на ринку.

Однією з ключових проблем, з якою стикаються користувачі 3D-друку, є якість виготовлених деталей, зокрема наявність дефектів, що можуть впливати на функціональні характеристики продукту. У процесі друку можуть виникати різні дефекти, такі як тріщини, порожнини або деформації, які не лише погіршують якість кінцевого продукту, а й призводять до перевитрат матеріалів і часу. Зменшення кількості дефектів під час 3D-друку не лише підвищує ефективність виробництва, але й дозволяє отримувати більш надійні та довговічні вироби, що є важливим для високотехнологічних галузей, де точність і якість мають критичне значення.

Для вирішення цієї проблеми дедалі більшого значення набувають програмні засоби для аналізу якості 3D-друку, розробка яких дозволяє автоматично виявляти дефекти на зображеннях готових виробів та проводити їх якісну оцінку. Завдяки цьому виробники можуть оперативно реагувати на проблеми, мінімізувати втрати, пов'язані з браком, та покращувати якість кінцевої продукції. Програмні інструменти для аналізу дефектів підвищують стандарти виробництва, забезпечують контроль за кожним етапом виготовлення і дозволяють компаніям дотримуватись вимог сучасних ринків, де висока якість є обов'язковою умовою.

Розвиток 3D технології сприятиме відродженню промисловості, надаючи вітчизняним підприємствам можливість виробляти конкурентоспроможні продукти з мінімальними витратами. Окрім того, завдяки 3D-друку та ефективному контролю якості можна створювати товари, необхідні для задоволення актуальних потреб країни, зокрема у військовій та медичній сферах. Це допоможе знизити залежність від імпорту та розвивати внутрішнє виробництво, що є стратегічно важливим для економічної безпеки та незалежності України.

Таким чином, впровадження програмного засобу для аналізу якості 3D-друку має вагомим значення як для підвищення ефективності виробництва, так і для зміцнення промислового потенціалу України в сучасних умовах.

Об'єкт дослідження – процес 3D-друку та якість виготовлення тривимірних виробів.

Предмет дослідження – методи автоматизованого аналізу зображень для виявлення дефектів у готових 3D-друкованих деталях.

Мета дослідження – розробка програмного засобу для аналізу зображень 3D-друкованих виробів з метою виявлення дефектів, що дозволить підвищити якість кінцевої продукції та мінімізувати втрати в процесі виробництва.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз видів технологій 3D-друку;
- проаналізувати властивості існуючих матеріалів для 3D-друку;
- на основі виконаного аналізу, обрати технологію та матеріали, які будуть використовуватися для виготовлення дослідних зразків для подальшого пошуку дефектів;
- дослідити інтерфейс, параметри та можливості програми-слайсера для підготовки 3D-моделі до друку;
- виконати обробку зображень деталей, виготовлених методом 3D-друку для подальшого пошуку дефектів за допомогою програмного засобу.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Види технологій 3D-друку

3D-друк є актуальним у сучасних виробничих та науково-дослідних процесах, оскільки забезпечує можливість швидкого прототипування та тестування виробів ще на етапі розробки. Це дозволяє компаніям та інженерам оперативно створювати дослідні моделі, оцінювати їх функціональність, виявляти потенційні недоліки та вносити корективи до кінцевого продукту без великих фінансових і часових затрат.

Основні переваги 3D-друку в цьому контексті:

- швидке створення прототипів, що дозволяє розробляти і тестувати ідеї в реальних умовах, знижуючи час на випуск продукту на ринок;
- гнучкість у модифікаціях, яка дає змогу швидко вносити зміни в конструкцію і перевіряти їх на практиці, що є складним і витратним процесом у традиційному виробництві;
- зменшення витрат на розробку за рахунок економії матеріалів і часу порівняно зі створенням моделей традиційними методами;
- оптимізація продукту, а саме завдяки швидкому доступу до дослідних виробів, компанії можуть покращувати конструкцію і досягати більш високої якості ще на етапі прототипування, що підвищує загальну якість і функціональність кінцевого продукту.

Таким чином, 3D-друк відкриває нові можливості для ефективного управління розробкою, скорочує цикл виробництва та забезпечує конкурентоспроможність у різних галузях, де важливо мати готовий продукт у найкоротші терміни.

Крім апаратного (hardware) аспекту, де основну роль відіграють принтери та механічні компоненти, в процесі друку задіяні різні аспекти. Можна також виділити

наступні підходи за технологією роботи та застосуванням:

- програмний (software) аспект охоплює програмне забезпечення для 3D-моделювання, підготовки моделей до друку (програми слайсери), а також програми для аналізу та оптимізації моделей. Важливі програми для діагностики дефектів, калібрування принтерів, контролю якості та моніторингу процесу друку. Наприклад, Autodesk Fusion 360, Cura, Simplify3D, PrusaSlicer та ін.;

- матеріальний зосереджується на розробці та вдосконаленні матеріалів для друку. Включає розробку нових видів філаментів, фотополімерів, порошків і паст, а також композитних і біологічних матеріалів, що відповідають різним вимогам до міцності, гнучкості, термостійкості та екологічності;

- процесний аспект охоплює методи і технології друку (як-от FDM, SLA, SLS тощо) та стандартизацію виробничих процесів. Він також включає розвиток нових підходів до створення багатокomпонентних або гібридних виробів, автоматизацію та інтеграцію 3D-друку в традиційні виробничі процеси;

- медичний та біологічний напрямок, спрямований на створення біологічних тканин, імплантів і протезів з використанням біосумісних або друкарських матеріалів;

- науково-дослідний аспект включає дослідження нових методів, матеріалів, програмного забезпечення та застосувань 3D-друку в різних галузях, веде до створення прототипів, тестування нових концепцій друку і вдосконалення наявних технологій;

- екологічний – зосереджений на розробці рішень для зменшення екологічного впливу 3D-друку, наприклад, використання біорозкладних матеріалів, технологій друку з переробленого пластику або розробка економічних принтерів, які споживають менше енергії.

Кожен з цих аспектів направлений на різні цілі і може комбінуватися для створення більш досконалих виробничих систем у сфері 3D-друку.

Технології друку можна розділити на кілька основних видів залежно від

принципів роботи. Далі розглянемо найпоширеніші з них:

– FDM (Fused Deposition Modeling) або FFF (Fused Filament Fabrication). Це найпоширеніша і найдоступніша технологія, за якої розплавлений пластиковий філамент подається через екструдер та шар за шаром наноситься для створення моделі. Використовує такі матеріали, як PLA, ABS, PETG та інші термопластики;

– SLA (Stereolithography). У цій технології застосовуються фотополімерні смоли, які застигають під дією ультрафіолетового (УФ) лазера, дозволяє досягти дуже високої точності та деталізації, що підходить для виготовлення прототипів і деталей з гладкою поверхнею;

– DLP (Digital Light Processing). Схожа з SLA, але замість лазера використовується цифровий проектор, який одночасно застигає цілий шар. Це значно пришвидшує процес друку і зберігає високу точність;

– SLS (Selective Laser Sintering). Ця технологія використовує лазер для спікання частинок порошкових матеріалів (зазвичай поліаміду або нейлону), шар за шаром створюючи об'єкт. SLS підходить для міцних і функціональних прототипів і не потребує опор, оскільки невикористаний порошок служить підтримкою для деталей;

– MJF (Multi Jet Fusion). Розроблена компанією HP, ця технологія також використовує порошкові матеріали, але замість лазера для їх спікання застосовуються спеціальні агенти та джерела тепла;

– PolyJet технологія, що працює за принципом струменевого друку, при якому рідкі фотополімери наносяться шар за шаром та відразу полімеризуються під дією УФ-ламп. PolyJet дозволяє використовувати кілька матеріалів одночасно, створюючи моделі з різними кольорами та властивостями;

– DMLS (Direct Metal Laser Sintering) і SLM (Selective Laser Melting). Обидві технології використовують лазер для створення металевих деталей з порошку, але різниця полягає в тому, що SLM повністю плавить порошок, а DMLS спікає його. Ці методи підходять для виробництва міцних металевих компонентів, які часто

використовуються в авіаційній і медичній промисловостях;

– EBM (Electron Beam Melting) схожа на SLM, але замість лазера використовується електронний промінь для плавлення металевого порошку. EBM підходить для деталей, що потребують високої міцності, і часто використовується в аерокосмічній та медичній галузях.

– LDM (Liquid Deposition Modeling) – це метод, за якого використовуються пастоподібні матеріали (наприклад, глина, кераміка або їстівні матеріали). Шар за шаром наноситься паста, яка затвердіває, формуючи кінцевий продукт. Цей вид 3D-друку набув популярності в мистецтві та кулінарії.

Кожна з цих технологій має свої унікальні характеристики і використовується залежно від вимог до матеріалу, точності, швидкості та бюджету на виробництво.

### 1.1.1 Типи матеріалів для 3D-друку

Далі розглянемо особливості деяких матеріалів, які використовуються в 3D-друці, їх призначення та порівняємо переваги та недоліки для того, щоб обрати матеріал для подальшого використання в дослідженні, таблиця 1.1.

Таблиця 1.1 – Переваги та недоліки основних типів матеріалів для 3D-друку

	Матеріал	Використання	Переваги	Недоліки
1	Пластик	Найпоширеніший матеріал для 3D-друку завдяки доступності та різноманітності варіантів. Використовується в прототипуванні, виготовленні функціональних виробів, а також у хобі та домашніх проектах.	Легкий, недорогий і доступний матеріал. Простий у друку, підходить для створення як функціональних деталей, так і декоративних елементів. Різноманітність видів пластику (PLA, ABS, PETG) для різних застосувань.	Обмежена міцність і термостійкість у порівнянні з іншими матеріалами (наприклад, металом). Деякі види пластику мають високе виділення токсичних парів (ABS).

Продовження табл. 1.1

2	Фотополімер	Використовується там, де потрібна висока деталізація та гладка поверхня, наприклад, у ювелірних виробках, стоматології та прототипуванні.	Висока точність і деталізація. Можливість створення складних структур з гладкою поверхнею. Підходить для моделей, що потребують специфічних деталей, як-от мініатюри або стоматологічні форми.	Вимагає УФ-обробки для затвердіння, що може підвищувати час виробництва. Смоли можуть бути токсичними і дорогими. Фотополімери зазвичай менш міцні, ніж інші матеріали, і мають обмежену стійкість до навантажень.
3	Метал	Популярний у високотехнологічних галузях, таких як авіація, автомобілебудування та медицина, для виробництва функціональних і міцних деталей.	Висока міцність і стійкість до зношування, придатний для механічно навантажених деталей. Стійкий до температурних змін і агресивних середовищ. Підходить для виготовлення складних деталей, які неможливо виготовити традиційними методами.	Висока вартість обладнання та матеріалів. Друк металом потребує спеціальних умов і захисту, наприклад, контроль атмосфери. Складність у постобробці.
4	Кераміка	Використовується у виробництві виробів, які мають бути стійкими до високих температур або хімічного впливу, наприклад, у медицині, мистецтві та промисловості.	Стійка до високих температур і корозії. Висока міцність і твердість. Підходить для деталей, які повинні бути хімічно інертними (наприклад, у біомедичній інженерії).	Вимагає додаткового обпалювання для затвердіння. Керамічні вироби можуть бути крихкими. Друк керамікою зазвичай займає більше часу і є технологічно складнішим.
5	Композити	Використовуються в інженерії та промисловості для створення міцних, легких і термостійких деталей.	Підвищена міцність завдяки армуванню (вуглеволокно, скловолокно). Стійкість до деформацій і підвищена довговічність. Підходить для створення функціональних деталей і прототипів, що витримують високі навантаження.	Дорожчі матеріали, ніж звичайний пластик. Друк композитами може вимагати спеціалізованого обладнання. Композитні вироби важко піддаються постобробці, що обмежує їх точність.

Продовження табл. 1.1

6	Біоматеріали	Використовуються в медицині для біодруку живих тканин, створення імплантів, а також у біотехнологічних дослідженнях.	Можливість друку живих тканин і навіть органів. Підходить для створення біосумісних імплантів. Використовується в регенеративній медицині та наукових дослідженнях.	Дуже висока вартість і складність технології. Вимагає стерильних умов і контролю навколишнього середовища. Складний процес біодруку, що потребує спеціальних знань і технологій.
7	Пісок та піщані суміші	Використовується для створення форм для лиття, архітектурних моделей і декоративних елементів.	Добре підходить для великих об'єктів, таких як ливарні форми. Можливість виготовлення складних геометрій для промислового застосування. Економічно ефективний при виготовленні ливарних форм.	Крихкість матеріалу обмежує його застосування. Складність у досягненні високої деталізації та гладкої поверхні. Обмежена міцність у порівнянні з металевими та пластиковими виробами.

Таким чином, ознайомившись з застосуванням різних матеріалів для друку, їх властивостей, переваг та недоліків, можна зробити висновок, що найкращим матеріалом для дослідження дефектів 3D-друку є пластик. Він дозволяє швидко створювати прототипи й тестувати різні варіанти деталей, що важливо для досліджень з метою зменшення дефектів у виробі. Пластик також має широкий спектр механічних властивостей, що дозволяє адаптувати його до різних функціональних вимог і точно налаштувати якість друку, забезпечуючи зручність у роботі та можливість оптимізації виробничого процесу.

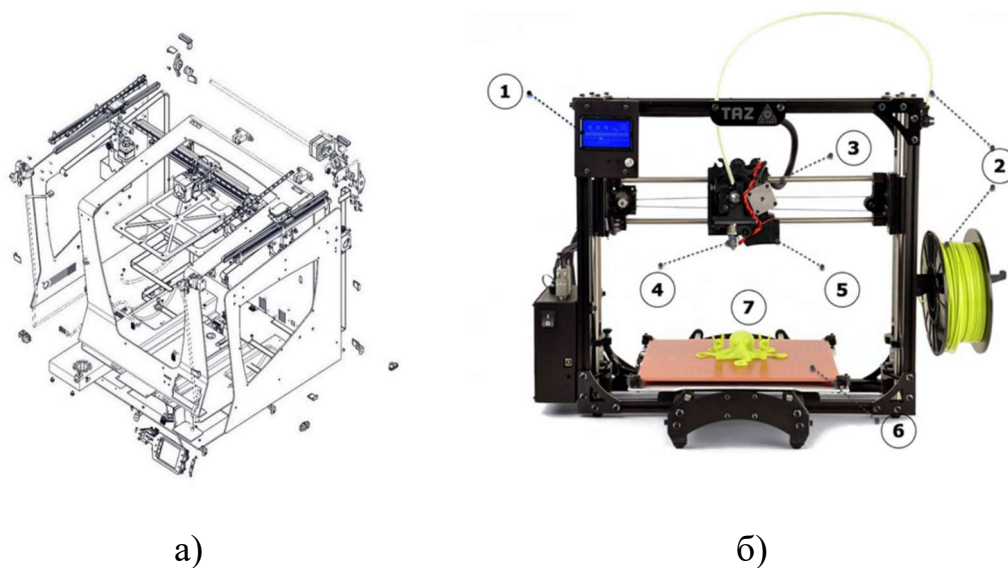
### 1.1.2 Схема побудови та компоненти 3D-принтера

Існує безліч 3D-принтерів за технологією FFF/FDM, різних конструкцій. Вони бувають різних розмірів і форм, але завжди містять наступні елементи:

- кінематика (система, що керує рухом механічних частин);

- екструдер;
- друкована платформа (стіл);
- різні електричні та електронні компоненти (наприклад, плата керування, блок живлення, датчики вирівнювання столу та кінця філаменту);
- допоміжні елементи (такі як рама, регулятори столу, тощо).

На рисунку 1.1 зображено загальну структурну схему побудови 3D-принтера.



а – загальна схема побудови 3D-принтера; б – схема побудови 3D-принтера для FDM технології.

1 – інтерфейс управління 3D-принтера; 2 – пластик для 3D-друку (філамент);  
3 – екструдер; 4 – хотенд (нагрітий кінець); 5 – вентилятор (кулер); 6 – друкована платформа (стіл); 7 – 3D-друк.

Рисунок 1.1 – Елементи 3D-принтера

Далі більш детально розглянемо компоненти принтера. 3D-принтер зазвичай складається з кількох ключових частин:

- каркас і платформа для друку, де каркас забезпечує стабільність, а платформа (або стіл для друку) слугує основою, на якій формується виріб;

- екструдер і хотенд (нагрітий кінець), де екструдер подає філамент, а хотенд плавить матеріал для нанесення шарів;
- драйвери двигунів і механізми руху. Крокові двигуни і рейки переміщують платформу та екструдер для точного формування шарів;
- система охолодження, де вентилятори охолоджують матеріал після екструзії для забезпечення стабільності форми;
- контролер – центральна плата, що керує всіма процесами, включаючи нагрів, рух і подачу матеріалу, часто з допомогою спеціалізованого програмного забезпечення або слайсера;
- електроживлення, а саме блок живлення, що постачає енергію для всіх компонентів.

## 1.2 FFF/FDM технологія 3D-друку

### 1.2.1 Основні відомості про FFF/FDM технологію

В цьому підрозділі розглянемо більш детально принцип роботи технології виготовлення методом наплавлення нитки або FFF. Fused Filament Fabrication (скорочено FFF) – це процес створення об'єкта шляхом пошарового нанесення розплавленого матеріалу. Нанесення контролюється спеціальною програмою (слайсером), що задає точну послідовність дій. Ця технологія використовує спеціальний вид матеріалу для 3D-друку, тобто філамент, який зазвичай виготовляється з пластику. Термін Fused Deposition Modeling (FDM) [8] часто використовують як синонім до FFF. Останній був створений розробниками проєкту RepRap, який з'явився у 2005 році.

На сьогодні FFF/FDM 3D-друк є більш поширеним і доступним, ніж будь-коли раніше. Завдяки проєкту RepRap і створеним ними FFF-пристроєм, загальні ціни на 3D-принтери значно знизилися. Тепер, коли вартість цих машин зменшилася приблизно в 10 разів, технологія стала доступною для великої кількості ентузіастів

та винахідників. Сучасна спільнота творців є значною, відкритою і постійно розвивається.

Після закінчення патентів компанії Stratasys на технологію FDM ринок заповнили якісні 3D-принтери з нагрівальними камерами, виготовлені іншими виробниками. З того часу терміни “FDM” та “FFF” стали практично синонімами. Сьогодні майже кожен винахідник та компанія використовують термін “FFF” для позначення 3D-принтера, що працює з матеріалами на основі пластику, навіть якщо він не оснащений нагрівальною камерою.

Зазвичай перші кроки в 3D-друці починаються з надійного і бюджетного FFF/FDM принтера. Проте адитивне виробництво має багато інших різновидів, таких як селективне лазерне спікання (SLS), стереолітографія (SLA), друк гіпсом і металом, а також інші технології. Однак найпопулярніший вибір для друку – це поки що настільний FFF/FDM 3D-принтер.

Основні етапи роботи цієї технології:

– підготовка 3D-моделі. Спочатку створюється цифрова 3D-модель об’єкта за допомогою спеціального програмного забезпечення (CAD-програми). Потім модель розбивається на шари в програмі-слайсері (наприклад, Cura, PrusaSlicer), що генерує G-код – набір команд для принтера;

– подача матеріалу. Друк відбувається за допомогою термопластичних матеріалів, таких як PLA, ABS, PETG. Філамент, який має вигляд пластикової нитки, подається в екструдер – механізм, що плавить матеріал;

– екструзія матеріалу. Екструдер розігріває філамент до температури плавлення, після чого матеріал витискається через гарячу форсунку. Сопло рухається по заздалегідь визначеному маршруту, щоб нанести матеріал у потрібних місцях;

– формування шарів. Друк відбувається пошарово. Коли один шар матеріалу охолоне, принтер починає наносити наступний. Цей процес повторюється, доки не буде сформовано весь об’єкт. Кожен шар з’єднується з попереднім, створюючи міцну структуру;

– завершення друку та постобробка. Після завершення друку об'єкт знімається з друкованої платформи. У деяких випадках необхідна додаткова обробка, така як видалення підтримок або шліфування поверхні для кращої якості.

Технологія FDM/FFF популярна завдяки своїй доступності, відносно низькій вартості матеріалів і простоті експлуатації.

Таким чином, комбінація з пластику та технології FFF/FDM є оптимальним вибором для даного дослідження з кількох причин.

По-перше, вони забезпечують економічність та доступність, оскільки FFF/FDM друк є одним з найбільш поширених і дешевих методів, а пластик (такі види як PLA, ABS, PETG) – відносно недорогий матеріал, що дозволяє виконувати експерименти з мінімальними витратами. Пластик також забезпечує швидке виготовлення прототипів, даючи змогу оперативно досліджувати пробні зразки та тестувати гіпотези, що є важливим у рамках дослідження.

Окрім того, пластикові матеріали широко використовуються в промисловості для прототипування, тому результати роботи з аналізу дефектів і оптимізації друку матимуть практичну цінність у різних інженерних проєктах. Технологія FFF/FDM полегшує ідентифікацію дефектів, таких як розшарування та проблеми з екструзією, що є особливо актуальним для розробки інструменту для аналізу якості друку.

Додатково, цей метод є відносно безпечним через мінімальну кількість шкідливих випарів, що спрощує роботу з експериментальними зразками. Таким чином, пластик і FFF/FDM найкраще відповідають вимогам даного дослідження, забезпечуючи практичність і безпеку роботи.

### 1.2.3 Принтери, які працюють з технологією FFF/FDM

Залежно від пристрою, процес 3D-друку виконується по-різному. Основною і найважливішою характеристикою, що лежить в основі кожного 3D-принтера, є його кінематика. Коротко кажучи, це система, яка контролює рухи екструдера.

Деякі компанії також використовують промислові маніпулятори, окрім звичайних

3D-принтерів. Однак існує широкий вибір і різноманітних роботизованих маніпуляторів для невеликих лабораторій, навчальних закладів, майстерень та інших суміжних застосувань. Відповідно до своєї кінематики, 3D-принтер може належати до однієї з наступних категорій.

Картезіанські 3D-принтери – принтери, які використовують кінематику картезіанського типу та є найпопулярнішими. Вони названі на честь Рене Декарта і базуються на координатній системі, яку він винайшов. Як правило, друкуюча головка таких машин рухається вздовж осей X і Y, тоді як друкована платформа переміщається вздовж осі Z, рисунок 1.2.

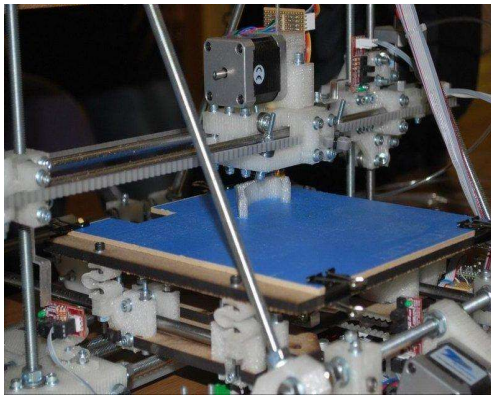


Рисунок 1.2 – Вигляд принтера картезіанського типу

Реалізація картезіанської кінематики може відрізнятися залежно від виробника. Наприклад, існують по-різному спроектовані 3D-принтери картезіанського типу, вироблені компаніями Prusa, Felix, Ultimaker, H-Bot, CoreXY та іншими виробниками.

Дельта 3D-принтер. Особливістю пристроїв дельта-типу є їхній екструдер. Він підвішений над друкованою платформою, утворюючи велику літеру  $\Delta$  (Дельта) грецького алфавіту. До речі, дельта 3D-принтери також працюють у картезіанській площині. Але, на відміну від картезіанських принтерів, дельта принтери використовують і полярну систему координат, яка застосовується під час процесу друку (рисунок 1.3).



Рисунок 1.3 – Вигляд принтера дельта-типу

Друкуюча головка принтера підтримується трьома рухомими важелями, які можуть швидко підніматися і опускатися вздовж стрижнів і рейок принтера. Така система руху значно підвищує швидкість друку дельта-принтерів, надаючи їм перевагу над картезіанськими моделями. Крім того, такі принтери оснащені круглими платформами для друку, які не можуть рухатися. Такий дизайн допомагає усунути небажані вібрації, що могли б вплинути на друк або навіть зруйнувати його.

Як правило, дельта-принтери можуть створювати моделі великої висоти. Однак ці моделі мають зменшені розміри вздовж осей X і Y порівняно з картезіанськими принтерами. Іншим недоліком дельта-принтерів є невисока точність друку, що можна вважати недоліком загальної конструкції цього типу 3D-принтерів. Враховуючи, що 3D-друк усе ще відносно молода технологія і постійно розвивається, існує висока ймовірність того, що згадані недоліки будуть незабаром усунені.

Полярний 3D-принтер (рисунок 1.4). Як стає зрозуміло із назви, система полярної кінематики базується на полярній системі координат. Друкуюча головка таких принтерів може рухатися лише вздовж осі Z (тобто підніматися або

опускатися), тоді як друкована платформа здатна обертатися і рухатися вздовж усіх осей, що дозволяє їй переміщатися ліворуч, праворуч, вгору і вниз.



Рисунок 1.4. – Вигляд принтера полярного виду

Пристрої полярного типу вирізняються безшумною роботою та компактним дизайном. Незважаючи на свою мініатюрність, вони мають потужну конструкцію, але займають більше місця, ніж Delta-пристрої.

Ці унікальні принтери не дуже популярні, тому для їхніх користувачів доступно менше онлайн-спільнот і програмних рішень.

SCARA (Selective Compliance Articulated Robot Arm) 3D-принтер складається з системи важелів, які переміщують робочу зону, обертаючи важелі відносно один одного. Пристрої на основі SCARA нагадують збиральні роботи, що використовуються в промислових додатках.

SCARA 3D-принтери оснащені точними приводами та з'єднувачами важелів, які коштують значну суму, що робить такі пристрої недоступними в умовах обмеженого бюджету (рисунок 1.5).

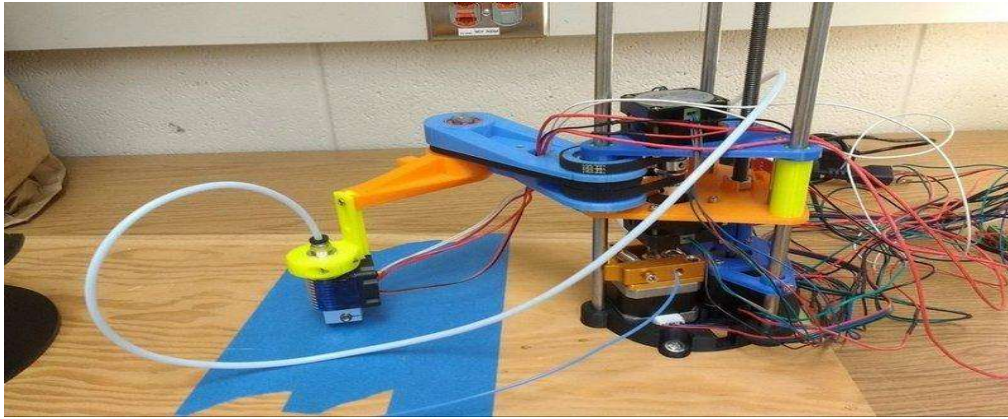


Рисунок 1.5 – Вигляд принтера виду SCARA

Враховуючи, що технології 3D-друку продовжують набирати популярність з кожним днем, це робить ринок 3D-принтерів більш конкурентоспроможним і знижує вартість таких машин. Сучасні 3D-принтери рідко оснащуються закритою камерою для друку чи навіть друкованою платформою і в основному розроблені для друку PLA. Деякі моделі, наприклад SkyOne, є універсальними і компактними модульними машинами, які можна навіть обладнати головкою для лазерного гравіювання.

### 1.3 Підготовка матеріалів для друку та налаштування параметрів принтера

Підготовка матеріалів та налаштування параметрів для друку на FFF/FDM 3D-принтері включає декілька етапів. Далі розглянемо окремо кожен із них.

Спершу підбирають матеріал у вигляді філаменту (пластикові нитка), враховуючи його сумісність із принтером і вимоги проєкту. Наприклад, PLA – легкий у друці, ABS стійкий до механічних впливів, а PETG є стійким до вологи.

Потім, філамент завантажують у принтер і підключають до екструдера. Перед друком виконують калібрування платформи, щоб забезпечити рівномірне нанесення матеріалу.

Наступний етап – налаштування параметрів друку в слайсері. Тут задають температуру екструдера й платформи, швидкість друку, товщину шару, заповнення

та інші параметри. Наприклад, PLA зазвичай друкується за температури від 190°C до 220°C, а ABS – від 220°C до 250°C. Товщина шару (від 0.1 до 0.3 мм) визначає деталізацію й міцність моделі. Чим тонший шар, тим вища якість, але довший час друку.

Після налаштувань запускається процес друку, під час якого принтер шар за шаром наносить розплавлений матеріал відповідно до створеної слайсером траєкторії.

### 1.3.1 Програми слайсери

Слайсер значною мірою визначає якість друку. Від правильного налаштування залежить точність, міцність, швидкість друку та якість поверхні готового виробу.

Слайсер – це програма, яка перетворює 3D-модель (зазвичай у форматах STL, OBJ або 3MF) на інструкції для 3D-принтера у вигляді G-коду. G-код містить покрокові інструкції, що вказують принтеру, як і де саме друкувати шар за шаром.

G-код (від англійського G-code) – це мова програмування, яка використовується для управління верстатами з числовим програмним управлінням (ЧПУ) та 3D-принтерами. У 3D друці G-код визначає послідовність дій, які має виконати принтер для створення об'єкта, включаючи рухи друкуючої головки, температуру, швидкість екструзії, швидкість охолодження тощо.

Основні функції G-коду:

- переміщення головки принтера, вказуючи на точні координати X, Y, і Z, куди потрібно перемістити друкуючу головку;
- контроль температури, тобто налаштувати температуру екструдера і нагрівальної платформи для правильного розплавлення матеріалу;
- налаштування швидкості, а саме встановлення швидкості подачі та руху, що впливає на якість і швидкість друку;
- керування екструзією, тобто контроль обсягу матеріала, який подається для створення шарів об'єкта.

Файли з G-кодом зазвичай генеруються програмами для слайсингу, як-от Cura, PrusaSlicer або Simplify3D, і використовуються для друку складених 3D-моделей, переводячи їх із формату STL у команди для принтера. Ці програми дозволяють готувати 3D моделі до друку та створювати G-код, який підходить для більшості 3D-принтерів.

Для редагування G-коду прямо можна використовувати такі інструменти [7]:

- Notepad++ – легкий текстовий редактор, що підтримує підсвічування синтаксису для G-коду, що полегшує редагування команд;
- NC Viewer – простий інструмент для візуалізації та редагування G-коду, призначений переважно для CNC, але підходить і для 3D друку;
- gCode Viewer – онлайн-редактор для візуалізації та редагування G-коду, розроблений спеціально для 3D друку.

Ці редактори дозволяють змінювати G-код для налаштування швидкості друку, руху та інших параметрів, залежно від потреб принтера та моделі.

Далі розглянемо основні функції програми-слайсера:

- розрізання моделі на шари. Слайсер «розрізає» модель на окремі горизонтальні шари, які принтер буде друкувати послідовно;
- налаштування параметрів друку. Дає можливість налаштувати параметри, такі як товщина шару, швидкість друку, температура нагріву екструдера та платформи, заповнення, підтримки, обрізка тощо;
- розрахунок траєкторії друку. На основі заданих параметрів програма визначає оптимальні шляхи, по яких буде рухатись екструдер для кожного шару;
- побудова підтримок. Деякі моделі потребують додаткових підтримок для друку складних частин. Слайсер може автоматично або вручну додавати ці підтримки.

Існує кілька популярних слайсерів, таких як:

- Ultimaker Cura – безкоштовний і дуже функціональний слайсер з відкритим кодом, підтримує різні принтери;

- PrusaSlicer – популярний серед користувачів Prusa та інших принтерів, зручний для налаштувань;
- Simplify3D – комерційний слайсер, відомий широким вибором параметрів та високою якістю підсумкового G-коду;
- ChiTuBox – часто використовується для фотополімерних принтерів (SLA/LCD/DLP).

### 1.3.2 Слайсер Ultimaker Cura

Ultimaker Cura – одна з найпопулярніших безкоштовних програм для слайсингу, яке перетворює 3D-моделі у G-код, що необхідний для друку. Ця програма підтримує різні формати файлів (STL, OBJ, 3MF) і сумісна з багатьма 3D-принтерами, не лише з бренду Ultimaker [10]. Вона пропонує простий інтерфейс для новачків та багато налаштувань для досвідчених користувачів.

На рисунку 1.6 зображено інтерфейс програми, розглянемо його елементи.

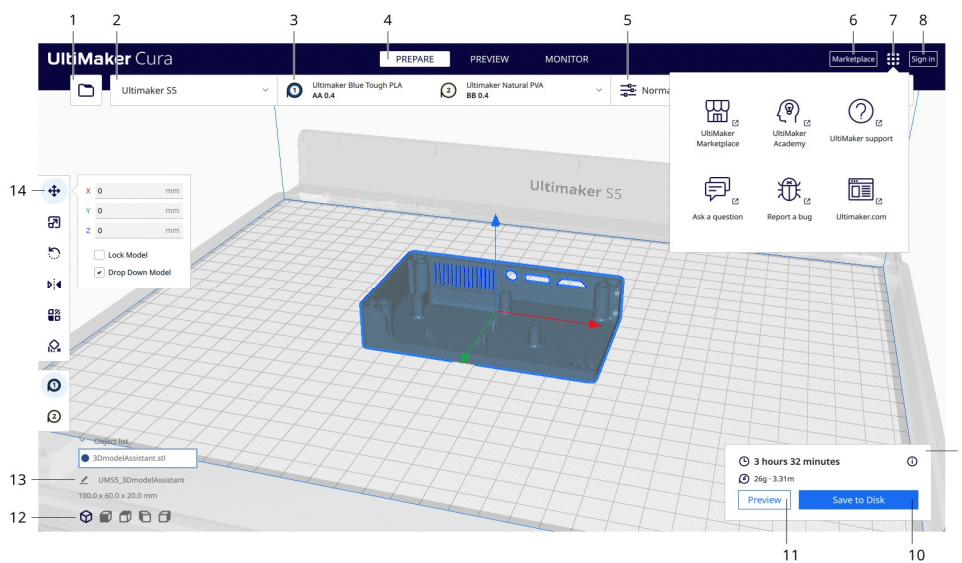


Рисунок 1.6 – Інтерфейс програми-слайсера Ultimaker Cura

Основними елементами слайсера є:

- кнопка «Відкрити файл», відкриває 3D файл;

- панель вибору принтера – відображає вибраний принтер. Тут можна активувати інший доданий принтер, додати нові принтери та керувати списком принтерів;
- панель конфігурації, що містить налаштування матеріалу та друкуючої голівки або сопла. Можна вручну вибрати іншу конфігурацію або обрати з доступних для підключених принтерів UltiMaker;
- етапи, які включають підготовку, попередній перегляд та моніторинг. Кожен етап структурований для ефективного виконання всіх кроків 3D друку;
- панель налаштувань друку, що містить профілі друку та всі індивідуальні налаштування друку;
- маркетплейс – містить плагіни та пакети матеріалів;
- перемикач додатків дозволяє легко перейти до додаткових платформ UltiMaker для підтримки, онлайн-навчання, тощо;
- обліковий запис UltiMaker. Вхід у обліковий запис UltiMaker, або створити новий для доступу до функцій хмари, плагінів маркетплейсу, профілів друку, резервних копій, тощо;
- панель дій містить кнопку дій, залежно від поточного етапу. Тут може бути кнопка для слайсингу моделі, а також інші опції після підготовки моделі;
- зберегти або поділитися файлом. Після слайсингу можна зберегти файл на комп'ютер або знімний носій, або ж запустити друк через локальну мережу. Якщо принтери UltiMaker підключені до Digital Factory, можна друкувати через хмару або зберегти проєкт у Digital Library;
- перехід до попереднього перегляду;
- інструмент позиції камери, що легко встановлює камеру на одну з основних точок огляду;
- інформація про модель містить назву завдання друку (на основі першої завантаженої моделі) та загальні розміри всіх друкованих моделей на платформі.

Якщо моделей декілька, вони відображаються у списку об'єктів;

- інструменти коригування включають інструменти для переміщення, обертання, дзеркального відображення, масштабування та інші опції для зміни вигляду чи орієнтації ваших 3D моделей.

Основні функції Cura включають:

- зручність у використанні з налаштуваннями для різних рівнів знань;
- профілі для друку, що оптимізовані для різних матеріалів;
- можливість детальної налаштування параметрів, як-от висота шару, щільність заповнення, швидкість друку тощо;
- плагіни для розширення можливостей програми;
- сумісність з різними принтерами, що дозволяє працювати навіть з обладнанням інших брендів.

Початок роботи включає завантаження моделі, налаштування параметрів друку, та генерацію G-коду. У Cura є режим попереднього перегляду, де можна бачити шари моделі для перевірки розташування підтримок і заповнення.

Ultimaker Cura допомагає запобігати дефектам друку, надаючи гнучкі налаштування, які дозволяють оптимізувати параметри друку залежно від конкретної моделі та матеріалу. Ось деякі способи, як ця програма може допомогти уникнути дефектів:

- налаштування висоти шару, а саме вибір оптимальної висоти шару дозволяє збалансувати якість та швидкість друку, зменшуючи ризик утворення тріщин або нерівностей;
- контроль швидкості друку та подачі. Занадто висока швидкість може призвести до дефектів, як-от недруковані ділянки або зсув шарів. Cura дозволяє налаштувати швидкість друку та екструзії, щоб забезпечити рівномірний потік матеріалу, що зменшує ризик таких дефектів;
- використання підтримок і заповнення. За допомогою цих налаштувань можна забезпечити стійкість моделей з виступами або порожнинами, що особливо

корисно для складних форм. Це запобігає обваленню виступів та зміщенню шарів;

- температурні налаштування – Cura дає можливість точно налаштувати температуру екструдера та столу, що дозволяє оптимізувати адгезію (прилипання) між шарами та уникнути таких дефектів, як викривлення країв чи погане зчеплення шарів;

- налаштування охолодження, тобто контроль вентиляторів охолодження дозволяє уникнути перегрівання матеріалу, особливо для тонких або складних деталей, що знижує ризик деформацій і зміщення.

Крім того, попередній перегляд друку в Cura дає змогу користувачу бачити, як кожен шар буде наноситися, що дозволяє виявити потенційні проблеми ще до початку друку.

#### 1.4 Дефекти 3D-друку

Дефекти в 3D-друці можуть виникати через кілька причин, пов'язаних з налаштуваннями принтера, якістю матеріалу, умовами середовища та конструктивними особливостями моделі. Найпоширеніші дефекти включають розшарування, деформацію, порожнини, нерівні шари та погане прилипання до платформи. Причинами дефектів може бути [4]:

- неправильно налаштована температура екструдера або платформи може призвести до поганого з'єднання шарів чи деформації;

- занадто висока швидкість може знижувати якість друку та точність, викликаючи появу дефектів на складних деталях;

- налаштування платформи – якщо платформа некалібрована, виріб може погано приклеїтися, що викликає деформацію чи відрив від поверхні;

- матеріал, а саме використання низькоякісного філаменту або невідповідного матеріалу підвищує ризик появи дефектів через нерівномірне розплавлення чи погане з'єднання шарів;

– конструкція моделі або невідповідний дизайн, наприклад відсутність опор для виступаючих частин, може призводити до зміщень або прогинів у моделі.

Умови довкілля, такі як температура повітря або вологість, також впливають на стабільність друку, особливо для пластиків, чутливих до коливань температур.

Далі розглянемо кожен тип дефекту окремо.

#### 1.4.1 Погана адгезія

Щоб отримати якісну модель, дуже важливо, щоб пластик добре прилип до платформи побудови (рисунок 1.7). Інакше виріб навряд чи продрукується належним чином.

Модель може погано прилипати до платформи з таких причин:

- неправильні налаштування системи охолодження;
- відсутність калібрування робочого столу;
- зазор між платформою та соплом занадто великий;
- температурний режим не відповідає типу матеріалу.

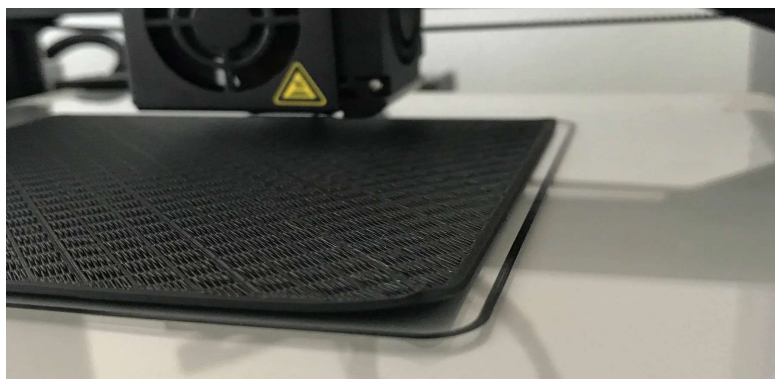


Рисунок 1.7 – Виріб з поганою адгезією

Якщо проблема криється в налаштуваннях, достатньо привести їх до вимог, що відповідають конкретному виду пластику. Також варто обов'язково виконати калібрування. У більшості пристроїв процес автоматизовано. Поліпшити адгезію здатні спеціальні матеріали – адгезійний спрей/клей, синій скотч, каптон тощо.

### 1.4.2 Надмірна, або недостатня екструзія

Дефекти моделей бувають спричинені збоями в подачі пластику (рисунок 1.8). Недоекструзія чи переоекструзія – проблеми протилежні, але причини у них ідентичні:

- неправильно заданий коефіцієнт екструзії;
- неправильно зазначений діаметр нитки;
- збої в механіці.

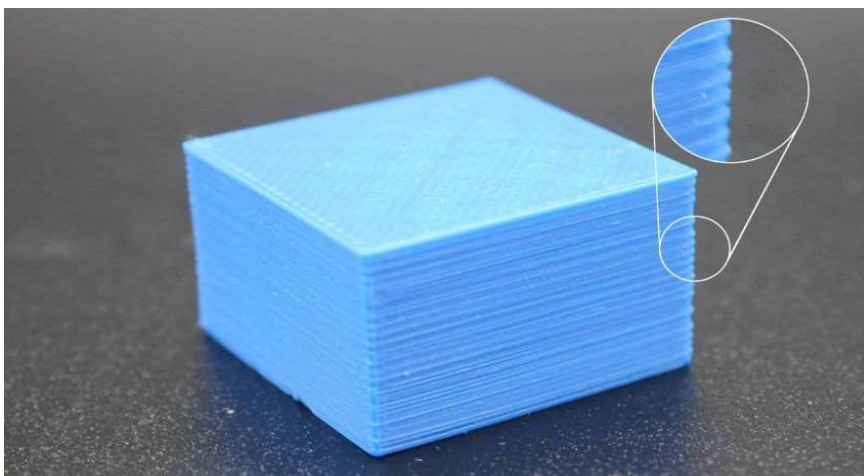


Рисунок 1.8 – Виріб з надмірною/недостатньою екструзією

В якості рішення пропонується переглянути налаштування та внести редагування. Також слід перевірити механізм притиску та блок, який відповідає за подачу пластику.

### 1.4.3 Надмірне нагрівання

Якщо пластик розігрівається до дуже високої температури і подається на «розпечену» платформу, то модель просто не встигатиме застигати. Логічно припустити, що підсумковий виріб навряд чи вийде таким, яким він замислювався. Проблема зазвичай зумовлена невідповідністю параметрів системи охолодження та швидкості друку. Вирішується вона приведенням налаштувань у норму. Важливо

розуміти, що у кожного типу пластика температура остигання та плавлення відрізняється (рисунок 1.9).

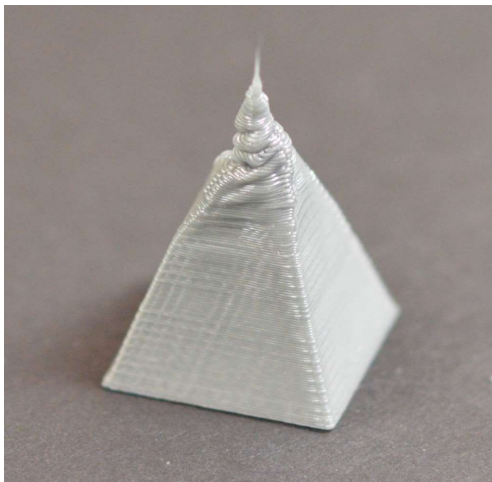


Рисунок 1.9 – Виріб з надмірним нагріванням

#### 1.4.4 Поява павутиння

Незначні павутинки витратного матеріалу в окремих точках об'єкта не становлять небезпеки (рисунок 1.10). Але якщо їх кількість велика, необхідно заглянути в налаштування.

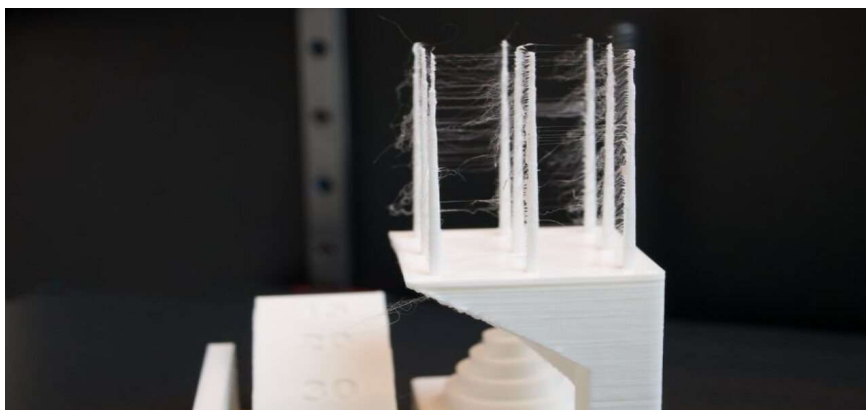


Рисунок 1.10 – Виріб з дефектом «павутинка»

Зазвичай виправити ситуацію дозволяє зміна характеристик втягування пластику. Значення, на які слід звернути увагу – швидкість втягування та дистанція.

Достатньо зробити пару проб, щоб зрозуміти, як правильно задавати відповідні параметри. Також проблема може ховатися у температурі. Велика температура може провокувати надмірне розплавлення матеріалу та його розтікання.

#### 1.4.5 Зміщення шарів

Ця проблема відноситься до категорії серйозних, адже найчастіше вона вказує на наявність проблем із механічними, електронними елементами 3D принтера (рисунок 1.11). Хоча можливий варіант, коли просто заданий занадто великий діапазон руху головки.



Рисунок 1.11 – Виріб зі зміщенням шарів

#### 1.4.6 Відсутність подачі пластику

Проблема подачі пластику (рисунок 1.12) зустрічається нерідко і відбувається найчастіше через:

- забите сопло (пробки);
- велику швидкість;
- низьку температуру.



Рисунок 1.12 – Виріб з пропущеними шарами

Спочатку можна спробувати виштовхнути матеріал вручну. Якщо цього недостатньо, потрібно перезавантажити пластик. Якщо маніпуляції не дали результату, тоді варто поекспериментувати із налаштуваннями. Або чищення сопла за допомогою відповідних голок.

#### 1.4.7 Розщеплення шарів, утворення тріщин

Іноді такі дефекти провокуються неякісним пластиком. Також серед можливих причин – надто високі шари, недостатня температура. Привести все в норму допоможуть правильні налаштування (рисунок 1.13).



Рисунок 1.13 – Виріб з розщепленням шарів

#### 1.4.8 Дефект «Слонова нога»

Це дефект, коли перші шари відбитка мають більшу ширину та виступають за габарити виробу. Виникає «слоняча нога», коли нижні шари не встигли охолонути і під тиском верхніх починають розплющуватися (рисунок 1.14).



Рисунок 1.14 – Виріб з дефектом «слонова нога»

Одна з основних причин – велика температура. Змінювати налаштування рекомендується поступово – з кроком приблизно 5 градусів. Якщо не включені обдування, то на перших шарах можна додатково активувати і його. Іноді «слонова нога» з'являється при недостатньому зазорі між платформою і соплом. У такому разі зайвий пластик витісняється.

#### 1.4.9 Дефект «Дзвін»

Дефект проявляється у вигляді «брижі» – тінь навколо елементів, які виступають «гостро». Утворюється дзвін у місцях, де вектор руху екструдера змінюється. Проблема може утримуватися в люфті екструдера, недостатньому натягу ременів. Також бриж часто з'являється через велику швидкість (рисунок 1.15).



Рисунок 1.15 – Виріб з дефектом «дзвін»

#### 1.4.10 Присутність щілин на моделях із тонкими стінками

Дефект передбачає наявність не тонкої цільної стінки, а 2 тонких стінок з наявністю між ними щілини (рисунок 1.16). Це може бути обумовлено невідповідністю діаметром сопла та заданого значення товщини стінки. Наприклад, сопло 0,4 мм. Щоб отримати стінку 1 мм, двох проходів буде недостатньо, а трьох проходів – забагато. Для вирішення проблеми слід доопрацювати 3D модель або скористатися слайсером з іншими алгоритмами. Сучасне програмне забезпечення постійно вдосконалюється, «вчиться» не допускати подібних недоліків, тому такі дефекти трапляються все рідше.



Рисунок 1.16 – Виріб зі щілинами на моделях із тонкими стінками

#### 1.4.11 Недотримання геометрії

Якщо з квадрата вийшов ромб, а з кола овал, то причина скоріш за все у механіці. Необхідно обов'язково перевірити ремені та шків.

Звичайно, це далеко не всі проблеми. Але в основному більшість вирішуються шляхом зміни налаштувань (рисунок 1.17).

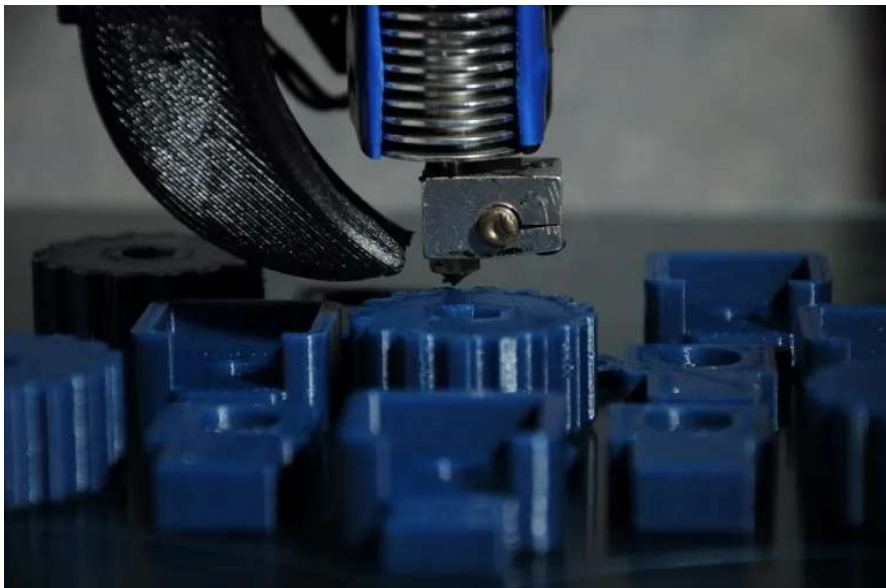


Рисунок 1.17 – Виріб з недотриманням геометрії

#### 1.5 Висновки до першого розділу

У результаті аналізу предметної області встановлено, що 3D-друк є сучасною технологією виробництва, яка дозволяє створювати складні деталі та моделі з використанням різноманітних матеріалів і методів. Розглянуто основні технології 3D-друку, включаючи FFF/FDM, які є одними з найпоширеніших завдяки доступності та простоті використання.

Досліджено ключові етапи процесу 3D-друку, зокрема підготовку матеріалів, налаштування параметрів принтера за допомогою слайсерів, таких як Ultimaker Cura, та вплив цих факторів на кінцеву якість друку. Особливу увагу приділено аналізу можливих дефектів, що можуть виникати під час друку, таких як погана адгезія,

надмірна екструзія, дефекти геометрії тощо. Встановлено, що більшість дефектів пов'язані з особливостями використаної технології друку, параметрами налаштування обладнання та властивостями матеріалів.

На основі проведеного аналізу зроблено висновок, що швидке й автоматизоване виявлення дефектів у 3D-друці є важливим елементом підвищення ефективності виробництва. Це дозволить оперативно усувати помилки, оптимізувати час виготовлення деталей та зменшити витрати на ресурси. Тому автоматизація контролю якості друкованих виробів є актуальним напрямком для подальших досліджень та впровадження.

## 2 МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ ДЛЯ ВИЯВЛЕННЯ ДЕФЕКТІВ У 3D-ДРУЦІ

Методи обробки зображень для виявлення дефектів у 3D-друці включають різноманітні технології, які допомагають визначати проблеми на різних стадіях друку. Основні методи [11] розглянемо нижче.

Комп'ютерний зір та штучні нейронні мережі (CNN). За допомогою камер і нейронних мереж можна відстежувати процес друку в реальному часі для ідентифікації таких дефектів, як ниткоподібні залишки (stringing) або деформація поверхні. CNN здатні аналізувати зображення і відрізнити дефекти від нормально надрукованих частин, використовуючи спеціально навчені моделі для автоматичної класифікації та коригування процесу друку у разі виявлення помилок. Цей метод дозволяє виявляти дефекти відразу після їх виникнення, що мінімізує відходи та покращує якість продукції.

Проекція світлових смуг (Fringe projection). Цей метод проектує світлові смуги на поверхню надрукованого об'єкта, що дозволяє визначати геометричні відхилення. Зміни у відбитому світлі вказують на дефекти, які можна виявити за допомогою аналізу зображення. Такий підхід використовується, наприклад, у друці зі складними формами, де потрібно точне визначення поверхневих аномалій.

Контроль якості на основі багатовидового огляду. Використовуючи кілька камер для зйомки з різних кутів, цей метод обробки зображень забезпечує виявлення дефектів у частинах, недоступних з одного ракурсу. Такий підхід дозволяє бачити всі сторони друкованого об'єкта та аналізувати структуру з різних точок, що підвищує точність контролю якості.

Алгоритми машинного навчання. Ці алгоритми дозволяють виявляти дефекти, аналізуючи патерни і повторювані структури на зображеннях. Машинне навчання особливо ефективно для складних випадків, коли дефекти можуть бути непомітними

для людського ока або дуже малими, наприклад, мікротріщини або неоднорідності у матеріалі.

Ці методи стають все більш доступними завдяки розвитку обчислювальної потужності та точності алгоритмів, що значно покращує якість друку та знижує витрати на виправлення дефектів на пізніх стадіях.

## 2.1 Обробка зображень з метою знаходження дефектів

### 2.1.1 Лінеаризація зображень

Лінеаризація виконується для нормалізації зображення для вирівнювання контрасту, що полегшує подальше виявлення недоліків. Розглянемо більш детально цей процес.

Лінеаризація зображення – це процес перетворення кольорового або градаційного зображення у форму, де інтенсивність пікселів зводиться до лінійного спектра. Цей метод використовується для усунення нерівномірностей у контрастності та яскравості зображення, що забезпечує стабільну базу для подальшої обробки та аналізу дефектів.

Процес лінеаризації часто починається із калібрування освітлення та корекції градацій зображення для вирівнювання контрасту. Зазвичай використовуються алгоритми корекції гамма, рівні сірого та адаптивні алгоритми для визначення рівномірного розподілу інтенсивностей пікселів. Деякі алгоритми лінеаризації, наприклад, застосовують методи на основі середніх значень, інші – гауссове розмиття для рівномірного розподілу пікселів по зображенню. Інструменти, які використовуються для лінеаризації, можуть включати такі бібліотеки, як OpenCV для Python або MATLAB з його набором для обробки зображень.

Результати лінеаризації дозволяють полегшити аналіз дефектів у 3D-друці, зокрема виявлення аномалій на поверхні об'єктів. Лінеаризоване зображення можна аналізувати за допомогою порогової обробки (бінаризації), де кожен піксель

виводиться як «чорний» або «білий» залежно від інтенсивності, щоб чіткіше виділити контури або дефекти. Цей підхід допомагає виявити, наприклад, мікротріщини чи розшарування на друкованому об'єкті.

Лінеаризацію зображення [14] за допомогою бібліотеки OpenCV виконують в тому числі і за допомогою функції:

```
edges = cv2.Canny(image_np, 100, 200)
```

А саме, застосування визначення країв, щоб підкреслити контури (використовуючи Canny Edge Detection для лінеаризації). Більш детально код, який виконує лінеаризацію, надано в Додатку Б.

Після виконання лінеаризації для подальшого виявлення дефектів застосовуються алгоритми порівняння аналогових точок та інші методи комп'ютерного зору, що аналізують лінійні розміри, відхилення та текстури поверхні. Наприклад, такі бібліотеки, як scikit-image у Python, можуть використовуватись для виконання структурного порівняння, що визначає будь-які відхилення від стандартної моделі, які можуть свідчити про дефекти друку.

Розглянемо ці процеси на прикладі надрукованої деталі.

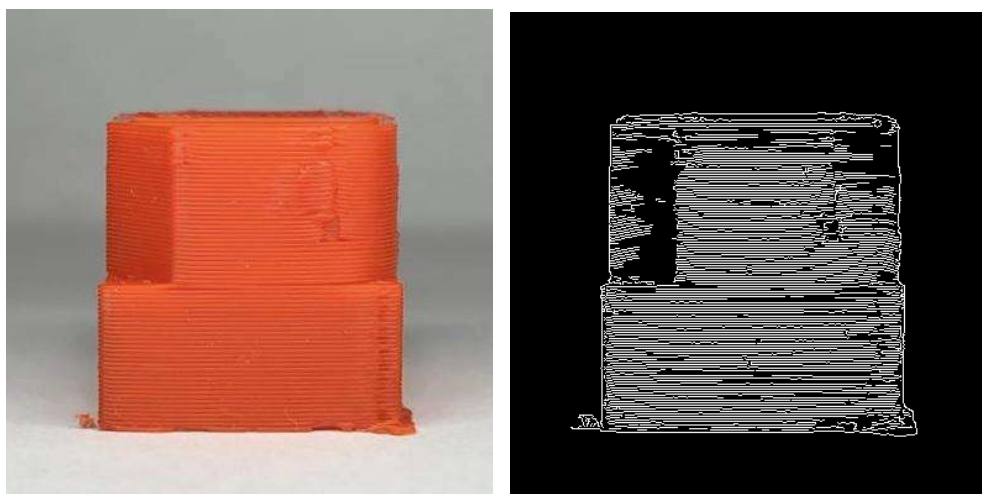


Рисунок 2.1 – Зображення до лінеаризації та після

Як видно на рисунку 2.1, проведена лінеаризація дозволяє побачити деталі структури та виявити можливі дефекти. Лінеаризація полягає у виділенні контурів та ліній на зображенні, щоб спростити аналіз і зробити дефекти більш помітними.

На цьому зображенні виділені контури, що можуть допомогти побачити деталі структури та виявити потенційні дефекти, наприклад, розшарування чи нерівності шарів.

### 2.1.2 Фільтрація

Фільтрація забезпечує видалення шумів за допомогою фільтрів (наприклад, Гаусового або медіанного) для чіткішого зображення об'єкту.

Далі розглянемо процес фільтрації більш детально. Фільтрація зображень – це процес обробки цифрових зображень, що дозволяє видалити шум, покращити якість зображення та виявити важливі особливості для подальшого аналізу. У контексті аналізу зображень 3D-друкованих деталей, фільтрація допомагає зосередитися на контрастних ділянках, підвищити чіткість контурів і виділити можливі дефекти, такі як нерівності шарів, тріщини чи інші артефакти, пов'язані з якістю друку.

Фільтрація може бути виконана за допомогою різних методів, залежно від мети. Основні типи фільтрів включають:

- середньозважений фільтр (Mean Filter) використовується для згладжування зображення та зменшення шуму. Він працює шляхом заміни кожного пікселя на середнє значення інтенсивностей пікселів навколо нього. Цей метод ефективний для видалення дрібного шуму, але може знизити чіткість деталей;

- медіанний фільтр (Median Filter), що також зменшує шум, особливо імпульсний (спеклінговий) шум. Він замінює значення пікселя на медіанне значення в його околі, що дозволяє зберегти контури на зображенні;

- гауссовий фільтр (Gaussian Filter) – згладжує зображення з меншим впливом на чіткі контури порівняно з середньозваженим фільтром. Застосовується для м'якого видалення шуму та покращення загальної якості зображення;

– лапласіанний фільтр (Laplacian Filter), що використовується для виявлення контурів та виділення різких змін інтенсивності. Допомагає краще побачити грані та межі об'єктів.

Для фільтрації зображень можна використовувати програмні засоби, такі як OpenCV, SciPy або інші бібліотеки для обробки зображень. Процес зазвичай передбачає вибір певного фільтру та налаштування параметрів, таких як радіус або маска згладжування.

Фільтрацію зображення за допомогою бібліотеки OpenCV виконують за допомогою функцій покращення країв фільтром Sobel, а саме `FIND_EDGES` та додаткового покращення контрасту для більшої чіткості. Детально код, який виконує фільтрацію подано в Додатку Б.

Після фільтрації отримане зображення може бути менш зашумленим і з чіткішими контурами, що допомагає алгоритмам обробки зображень точніше виявляти дефекти. У випадку з 3D-друкованими деталями, результати фільтрації дозволяють більш ефективно проводити наступні етапи обробки, такі як лінеаризація та бінаризація.

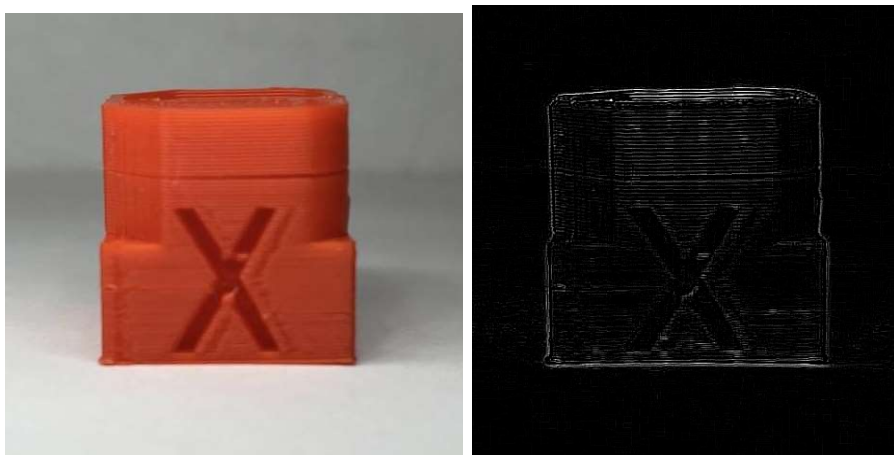


Рисунок 2.2 – Зображення до фільтрації та після

На рисунку 2.2 видно зображення після застосування фільтра Собеля, на якому чітко виділені контури об'єкта (3D-модель). Різкі зміни яскравості – це місця, де

текстура або поверхня змінюється, наприклад, на краях або рельєфних частинах моделі. Фільтр підкреслив ці контури, зробивши їх більш видимими, і також допоміг акцентувати увагу на дрібних деталях і можливих дефектах поверхні.

Після лінеаризації та фільтрації зображення стає більш придатним для алгоритмів виявлення дефектів, які можуть аналізувати чітко виділені контури, визначати точність розмірів і знаходити відхилення. Оброблені зображення можуть бути використані для створення карт дефектів, що допомагає визначити, які саме проблеми виникають під час 3D-друку.

### 2.1.3 Згладжування

Згладжування виконується для усунення дрібних нерівностей зображення, що допомагає уникнути хибного визначення дефектів.

Згладжування – це процес, який допомагає зменшити високочастотний шум, що може заважати точному розпізнаванню форм або дефектів на зображенні [14]. Наприклад, при 3D-друці дефекти можуть проявлятися у вигляді дрібних нерівностей або артефактів на поверхні, і завдяки згладжуванню такі артефакти стають менш помітними або їх можна краще ізолювати для подальшого аналізу.

Процес згладжування зображення є важливою операцією у обробці зображень, оскільки він дозволяє зменшити шум та інші дрібні артефакти, які можуть спотворювати аналіз об'єкта, що зображений. Згладжування спрямоване на те, щоб зробити перехід між сусідніми пікселями більш поступовим, таким чином зменшуючи різкість змін інтенсивності, але при цьому зберігаючи основні форми об'єктів.

Згладжування зазвичай досягається шляхом застосування різних фільтрів, наприклад таких як:

- гауссовий фільтр;
- медіанний фільтр;
- фільтр середнього значення, який замінює кожен піксель середнім

значенням інтенсивності його сусідніх пікселів, що зменшує варіативність інтенсивностей.

У сучасних програмах та бібліотеках для обробки зображень, таких як OpenCV, MATLAB, PIL (Python Imaging Library), є вбудовані функції для застосування фільтрів, включаючи згладжування. Наприклад, у OpenCV можна використовувати GaussianBlur, medianBlur, blur та інші методи для досягнення потрібного ефекту.

Згладжування зображення за допомогою бібліотеки OpenCV виконують за допомогою наступної функції:

```
smoothed_image = original_image.filter(ImageFilter.GaussianBlur(radius=2))
```

А саме, застосування розмиття за Гаусом для згладжування. Детально код, який виконує згладжування надано в Додатку Б.

Згладжене зображення з меншим рівнем шуму і плавнішими контурами допомагає краще виділяти основні дефекти, такі як нерівності поверхні або відхилення форми деталей, надрукованих на 3D-принтері. Після згладжування можна використовувати методи лінеаризації або інші техніки виділення контурів, щоб краще ізолювати ці дефекти для детального аналізу.

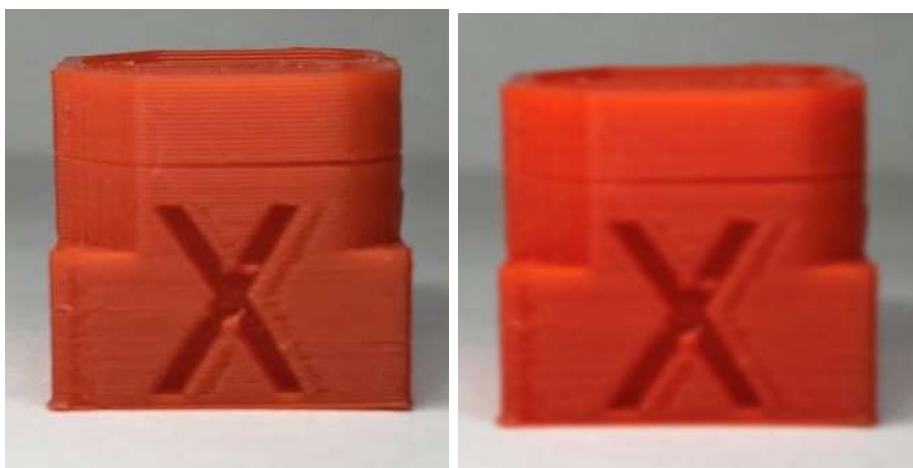


Рисунок 2.3 – Зображення до згладжування та після

Як видно на рисунку 2.3, після згладжування зображення стає менш «зашумленим» і готовим для подальшого оброблення, наприклад, для бінаризації, виділення контурів або інших методів сегментації, що дозволяють чіткіше ідентифікувати потенційні дефекти на поверхні надрукованих деталей. Згладжене зображення допомагає уникнути хибнопозитивних дефектів, які могли б виникнути через дрібні шуми або текстурні артефакти.

Використання таких методів у комплексі дозволяє точно ідентифікувати різні види дефектів, як-от тріщини, розшарування або деформації, що виникають в процесі друку. Це особливо корисно для налаштування параметрів друку та покращення якості виробів у FFF/FDM технології.

## 2.2 Висновки до другого розділу

У даному розділі було розглянуто ключові методи обробки зображень, зокрема лінеаризацію, фільтрацію та згладжування, які є важливими етапами для ефективного виявлення дефектів у деталях надрукованих за допомогою FFF/FMD технології.

Проведені дослідження показали, що:

- лінеаризація дозволяє перетворити зображення в зручний для аналізу формат, забезпечуючи однорідність даних та зменшуючи вплив нерівномірного освітлення або колірних викривлень;
- фільтрація допомагає виділити важливі деталі зображення, зокрема контури або дефекти, водночас усуваючи фонові шуми, що покращує точність подальшого аналізу;
- згладжування сприяє усуненню незначних локальних викривлень, які можуть виникати через оптичні спотворення чи неточності обладнання.

Ці методи забезпечують надійну основу алгоритму, який буде реалізовано в програмному засобі для контролю якості 3D-друкованих виробів. Їх поєднання

дозволяє мінімізувати вплив будь-яких викривлень на результати дослідження, що, своєю чергою, підвищує точність та об'єктивність виявлення дефектів.

Таким чином, обрані методи обробки зображень є критично важливими для функціонування програмного забезпечення, надаючи високу якість аналізу та оптимізацію процесу контролю виробництва.

## 3 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ

### 3.1 Бібліотека OpenCV

Для побудови програми, яка буде виконувати автоматичний пошук дефектів у 3D-друкованих деталях, необхідно використовувати інструмент, який дозволяє виконати наступні дії:

- обробляти зображення високої якості;
- порівняння двох зображень (еталонного та тестового) для виявлення відмінностей;
- автоматизувати процес аналізу зображень із високою точністю.

Одним з таких потужних інструментів є бібліотека OpenCV, що відповідає вимогам даного дослідження завдяки своїй багатофункціональності, продуктивності, простоті використання і можливостям роботи з графікою та обробкою зображень.

Також існують і інші інструменти, що дозволяють вирішувати подібні задачі, такі як Pillow (Python Imaging Library), scikit-image, MATLAB, Deep Learning Frameworks (наприклад, TensorFlow, PyTorch), ImageMagic та ін. Але всі вони мають свої обмеження та недоліки, що не дозволяють ефективно їх використовувати для цілей даного дослідження. Наприклад, вони мають обмежену функціональність для складних задач, вимагають значних ресурсів і часу для навчання моделей, або мають платний доступ та низьку продуктивність.

OpenCV створена за допомогою C++ і оптимізована для високошвидкісної обробки зображень, що є важливим чинником при роботі з великою кількістю даних. OpenCV має великий набір функцій для обробки та аналізу зображень, таких як вирівнювання, виявлення контурів, пошук ключових точок, а також алгоритми для аналізу текстур і кольорів. Бібліотека OpenCV легко інтегрується з іншими бібліотеками, такими як TensorFlow, PyTorch або NumPy, що дозволяє комбінувати

алгоритми комп'ютерного зору з машинним навчанням, а в подальшому дозволить розвивати продукт, розроблений в рамках даного дослідження.

В той самий час OpenCV підтримує інтеграцію також з Python та Java, що робить її зручним інструментом для швидкого прототипування, забезпечуючи високу продуктивність у фінальних рішеннях. OpenCV є безкоштовною та має відкритий код, на відміну від MATLAB. Також велика спільнота розробників та потужна документація роблять OpenCV доступним інструментом для вирішення складних задач.

Таким чином, для досягнення цілі даного дослідження, а саме розробки програмного засобу для пошуку дефектів у 3D деталях за допомогою порівняння зображень, OpenCV має такі переваги:

- дозволяє вирівнювання еталонного і тестового зображень для точного виявлення відмінностей (дефектів);
- надає можливість знаходити навіть дрібні дефекти за рахунок підтримки алгоритмів, таких як порівняння гістограм, методів ключових точок (ORB, SIFT);
- OpenCV дозволяє обробляти великі обсяги зображень у реальному часі, що підходить для автоматизації виробничих процесів;
- використання OpenCV в сукупності з Python дозволяє швидко створити робочий прототип і перевірити його на практиці, а також спрощує подальшу підтримку та можливість розвитку та модифікацій.

Таким чином, OpenCV забезпечує ідеальний баланс між функціональністю, продуктивністю та простотою використання, що робить її найкращим вибором для вирішення подібних завдань.

### 3.2 Аналіз мов програмування, які працюють з бібліотекою OpenCV

Згідно дослідження популярності мов програмування за використанням у 2024 році [16], найпоширенішими мовами, за допомогою яких можна побудувати додаток

та інтегрувати бібліотеку OpenCV Python займає перше місце, Java друге, а сама C++ – останнє (рисунок 3.2).

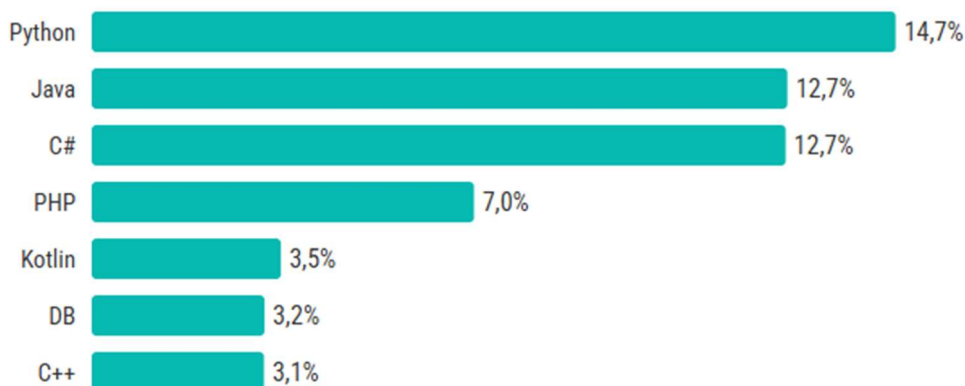


Рисунок 3.2 – Рейтинг популярності мов програмування

Розглянемо ці мови більш детально з точки зору використання в розробці програмного засобу для контролю якості друкованих виробів.

### 3.2.1 Мова програмування Java

Java – це об'єктно-орієнтована мова програмування, що широко використовується для розробки кросплатформених програм. Вона була розроблена в 1995 році компанією Sun Microsystems, яка нині належить Oracle. Основною ідеєю мови Java є «Write Once, Run Anywhere» (WORA), що означає, що код, написаний на Java, можна виконувати на будь-якій платформі, яка має Java Virtual Machine.

Основними характеристиками мови Java можна вважати платформну незалежність, об'єктну орієнтованість, надійність та безпеку, продуктивність та багатопоточність. Java має безліч бібліотек і фреймворків, таких як Spring, Hibernate, Apache Maven, які полегшують розробку великих проєктів.

Перевагами мови Java можна вважати:

- кросплатформеність, так як дозволяє розробникам створювати програми, які можуть працювати на будь-якому пристрої з JVM;

- масштабованість, тобто дана мова добре підходить як для невеликих додатків, так і для великих корпоративних систем;
- широка підтримка, а саме велика спільнота, постійне оновлення мови й фреймворків, а також потужна документація роблять Java легкодоступною для вивчення і розробки;
- можливість вирішення складних завдань. Java активно використовується для створення складних додатків у фінансовій сфері, інтернет-магазинах, мобільних додатках (особливо для Android), серверних додатках і системах великих даних.

Але так само, до недоліків Java можна віднести:

- відносно високу складність синтаксису. У порівнянні, наприклад, із мовою Python, Java є громіздкою через необхідність писати більше коду;
- використання великої кількості пам'яті. Програми на Java зазвичай споживають більше пам'яті, ніж програми, написані на C++ або Python, що вимагатиме більш дорогавартісного обладнання.

Також важливим аспектом у виборі мови програмування для даного дослідження є можливість інтеграції з бібліотекою OpenCV [15]. OpenCV підтримує Java API, що дозволяє використовувати її функції для обробки зображень і відео. Це зручно, якщо додаток потребує роботи з графічним інтерфейсом або є частиною мобільного додатку на Android. Java добре також підходить для завдань, які вимагають інтеграції комп'ютерного зору з іншими компонентами системи, а саме серверною обробкою зображень, розробкою Android-додатків та побудовою масштабованих систем.

Таким чином, Java є потужною мовою для розробки складних та масштабованих систем. Вона є складнішою за C++ або Python, що може уповільнювати розробку та подальшу підтримку програми, розробленої за допомогою Java за рахунок потреби в навчанні для спеціалістів, або збільшувати бюджет проєкта через потребу найма спеціаліста саме з цієї мови. Її використання виправдане для задач з комп'ютерного зору в мобільних додатках (Android) або

великих серверних рішеннях, але для швидкого прототипування, що є основною ціллю даного дослідження, доцільніше обрати простішу мову.

### 3.2.2 Мова програмування C++

C++ – це потужна мова програмування загального призначення, яка підтримує кілька парадигм програмування, зокрема об'єктно-орієнтовану, процедурну та узагальнену. Вона була розроблена Б'ярном Страуструпом у 1979 році в AT&T Bell Laboratories як розширення мови C, спочатку під назвою "C з класами". У 1984 році мова була перейменована на C++.

Основними характеристиками C++ є:

- підтримка різних стилів програмування, включаючи процедурне, об'єктно-орієнтоване та узагальнене програмування, що дозволяє розробникам вибирати найбільш підходящий підхід для конкретних завдань;
- забезпечення високої швидкості виконання програм, що робить її ідеальною для розробки програмного забезпечення, де критична ефективність, таких як ігри, системне програмування та обробка великих обсягів даних;
- підтримка низькорівневого програмування за рахунок того, що C++ дозволяє працювати з пам'яттю безпосередньо, що дає змогу розробникам оптимізувати використання ресурсів та досягати високої ефективності;
- наявність багатого набору стандартних бібліотек, які надають готові рішення для широкого спектра завдань, включаючи роботу з контейнерами, алгоритмами вводу/виводу та іншими аспектами програмування.

Враховуючи, що бібліотека OpenCV створена за допомогою C++ і вона є основною мовою для OpenCV та більшість її функцій реалізовані саме на C++, це забезпечує високу продуктивність та ефективність при розробці додатків для комп'ютерного зору. Однак, C++ має і свої недоліки. Наприклад:

- складність синтаксису, що може ускладнити навчання та розробку, особливо для початківців;

- C++ вимагає від розробника ручного управління пам'яттю, що може призвести до помилок, таких як витоки пам'яті, якщо не дотримуватися обережності;
- відсутність вбудованого механізму збору сміття. На відміну від деяких інших мов, C++ не має вбудованого механізму збору ресурсів, які не використовуються, що вимагає від розробника більшої уваги до управління ними.

Таким чином, C++ є потужною та гнучкою мовою програмування, яка підходить для розробки високопродуктивних додатків, включаючи ті, що використовують OpenCV для комп'ютерного зору. Однак, через свою складність, вона може бути менш зручною для швидкого прототипування порівняно з іншими мовами, такими як, наприклад, Python.

### 3.2.3 Мова програмування Python

Python – це високорівнева мова програмування загального призначення, яка була створена Гвідо ван Россумом у 1991 році. Її основна ідея – забезпечити читабельність і простоту коду, що робить її однією з найзручніших мов для розробників. Python підтримує кілька парадигм програмування, включаючи об'єктно-орієнтовану, процедурну та функціональну, що робить її універсальним інструментом для різноманітних завдань.

Основні характеристики Python:

- простота використання за рахунок зрозумілого синтаксиса, що нагадує англійську мову, завдяки чому його легко вивчати й використовувати навіть новачкам;
- кросплатформеність, тому що Python можна використовувати на різних операційних системах, таких як Windows, macOS і Linux, без змін у кодї;
- величезна стандартна бібліотека Python пропонує безліч готових модулів і функцій для роботи з текстом, числовими розрахунками, базами даних, мережею, обробкою зображень тощо;
- розширюваність – Python легко інтегрується з іншими мовами

програмування, такими як C, C++ та Java;

- велика кількість розробників, документація, приклади та ресурси роблять Python доступним для навчання й використання.

Python якнайкраще підходить для розробки програми для пошуку дефектів у 3D деталях завдяки таким перевагам:

- простота й швидкість прототипування Python дозволяє створювати й тестувати прототипи набагато швидше, ніж мови з більш складним синтаксисом, як C++ чи Java. Це важливо для дослідження та швидкої перевірки ідей;

- сумісності бібліотеки OpenCV та Python [19]. Використання OpenCV на Python є зручним і інтуїтивно зрозумілим, а сама бібліотека містить функції для обробки зображень, знаходження контурів, аналізу об'єктів та багато іншого;

- можливість подальшої інтеграції зі штучним інтелектом. Python має багатий набір бібліотек для машинного навчання, таких як TensorFlow, PyTorch, scikit-learn, які можуть бути використані для створення моделей, здатних автоматично знаходити дефекти на основі даних;

- легке навчання, тобто завдяки простоті Python його можуть використовувати навіть спеціалісти з обмеженим досвідом у програмуванні, що робить розробку доступнішою для міждисциплінарних команд.

Хоча Python і має певні недоліки, такі як швидкість виконання або використання великої кількості ресурсів, вона найбільше відповідає вимогам швидкої розробки, тестування та налаштування програми для пошуку дефектів у 3D деталях завдяки її простоті, багатій екосистемі бібліотек, а також широким можливостям інтеграції з OpenCV.

### 3.3 Розробка програми на Python та OpenCV

#### 3.3.1 Розробка основного алгоритму програми

Розглянемо етапи розробки програмного рішення. На перших етапах

виконується підключення бібліотеки OpenCV, обираються файли зображень та обчислюється MD5-хешу файлу для подальшої перевірки, що обрані різні файли. Нижче наведено код програми:

```
import cv2
import hashlib
import numpy as np
from tkinter import Tk, Button, Label, filedialog, messagebox
```

Останній рядок імпортує необхідні компоненти бібліотеки Tkinter, які забезпечують функціональність графічного інтерфейсу програми. Tk – це основний клас, який створює головне вікно програми та використовується для ініціалізації графічного інтерфейсу. Більше детально розробку графічного інтерфейсу розглянуто в підрозділі 3.3.2.

Далі наведені функції для виконання порівняння зображень, обчислення абсолютної різниці, перетворення різниці у чорно-біле зображення для подальшого аналізу, розрахунок кількості «білих пікселів», тобто відмінностей та обчислення відсотку схожості вже оброблених зображень. Наприклад: функція порівняння зображень: `def compare_images(image1, image2).`

Для того, щоб програма взаємодіяла із зображеннями та користувачем, далі запускається основна функція застосунка:

```
def main():
    root = Tk()
    root.title("Інструмент пошуку дефектів").
```

Детальніше функцію програми для пошуку дефектів надано в Додатку Б.

Також, програма виконує перевірку на дублікати файлів виконується двома способами – за шляхом: тобто якщо обрати один і той самий файл як еталонне зображення і як тестове, а також за розрахунком MD5-хешу файлу з зображенням, що

не пропустить далі навіть якщо той самий файл лежить в іншій папці та під іншою назвою. Наприклад:

```
if reference_path == new_path:
    messagebox.showerror("Помилка", "Однакові файли. Оберіть інший файл для порівняння.")
```

Таким чином, було розроблено алгоритм програми (Додаток Б), яка завантажує еталонне та тестове зображення, перевіряє, що вибрані різні файли, виконує обробку зображень з метою більш якісного їх порівняння та показує результат виконання порівняння – чи має тестова деталь дефекти в порівнянні з еталонною та відсоток схожості деталей.

### 3.3.2 Розробка інтерфейсу програми

Важливим аспектом розробки ПЗ є зовнішній вигляд, а також зрозумілість і простота використання. Дуже важливо взаємодіяти з користувачем, надавати зрозумілі підказки, а також чітко повідомляти про помилки та результати роботи. Зазвичай інтерфейс в Python будується за допомогою стандартної бібліотеки Tkinter. Це бібліотека для створення графічного інтерфейсу користувача (GUI), яка встановлюється разом із Python та дозволяє створювати прості та зручні вікна для взаємодії з користувачем. Наприклад, кнопка для запуску порівняння зображень в інтерфейсі будується наступним чином:

```
compare_button = Button(
    root,
    text="Оберіть зображення",
    font=("Arial", 12),
    bg="#006400", # Зелений фон
    fg="white", # Білий текст
```

```

activebackground="#228B22", # Темно-зелений при натисканні
height=2,
width=20,
command=on_compare # Виклик функції порівняння
)
compare_button.pack(pady=20)

```

Детально створення інтерфейса описане в Додатку Б.

### 3.3.3 Подальше розповсюдження ПЗ

За рахунок використання простих та доступних інструментів таких як Python та OpenCV, ПЗ, побудоване на їх основі, є простим в розповсюдженні та не потребує спеціальних навичок, що дозволяє оптимізувати витрати виробництва.

Для того щоб перемістити розроблену програму на інші комп'ютери, необхідно виконати наступні дії в наведеному порядку:

- створюємо виконуваний файл (.exe для Windows). Це дозволить запускати програму без необхідності встановлення Python;
- встановлюємо `pyinstaller` виконавши команду `pip install pyinstaller` в командному рядку. Успішне встановлення виглядатиме як показано на рисунку 3.3;

```

Downloading pyinstaller-6.11.1-py3-none-win_amd64.whl (1.3 MB)
 1.3/1.3 MB 13.6 MB/s eta 0:00:00
Downloading packaging-24.2-py3-none-any.whl (65 kB)
Downloading pefile-2023.2.7-py3-none-any.whl (71 kB)
Downloading pyinstaller_hooks_contrib-2025.0-py3-none-any.whl (344 kB)
Downloading pywin32-ctypes-0.2.3-py3-none-any.whl (30 kB)
Downloading setuptools-75.8.0-py3-none-any.whl (1.2 MB)
 1.2/1.2 MB 20.6 MB/s eta 0:00:00
Downloading altgraph-0.17.4-py2.py3-none-any.whl (21 kB)
Installing collected packages: altgraph, setuptools, pywin32-ctypes, pefile, packaging, pyinstaller-hooks-contrib, pyinstaller
Successfully installed altgraph-0.17.4 packaging-24.2 pefile-2023.2.7 pyinstaller-6.11.1 pyinstaller-hooks-contrib-2025.0
pywin32-ctypes-0.2.3 setuptools-75.8.0

```

Рисунок 3.3 – Повідомлення про успішне встановлення `pyinstaller`

- генеруємо виконуваний файл. Для цього необхідно перейти до папки з Python програмою та виконати команду `pyinstaller --onefile defect_identifier.py`. Де

команда `--onefile` створює один виконуваний файл, а `defect_identifier.py` визначає файл з програмою. Якщо створення `.exe` виконалось успішно, в командному рядку буде повідомлення: «115505 INFO: Building EXE from EXE-00.toc completed successfully.». Результат можна буде знайти в «dist» (рисунок 3.4);

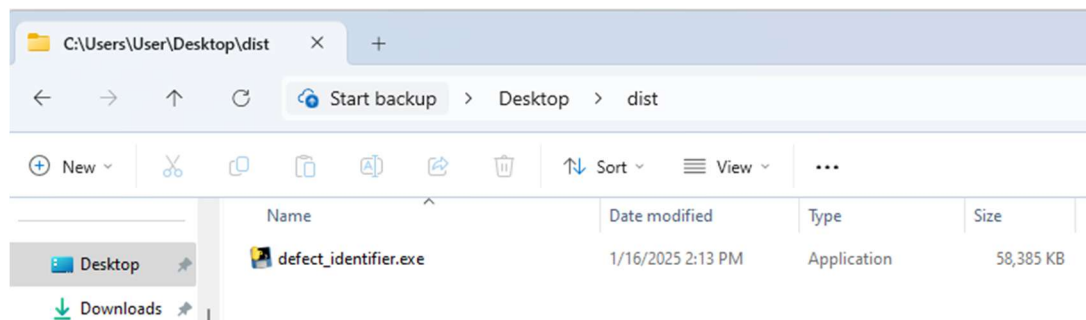


Рисунок 3.4 – Створений `.exe` файл для запуску програми на Windows

– перевіряємо виконуваний файл – знаходимо файл у папці `dist` і запускаємо його на комп’ютері без Python. Результат має бути як показано на рисунку 3.5.

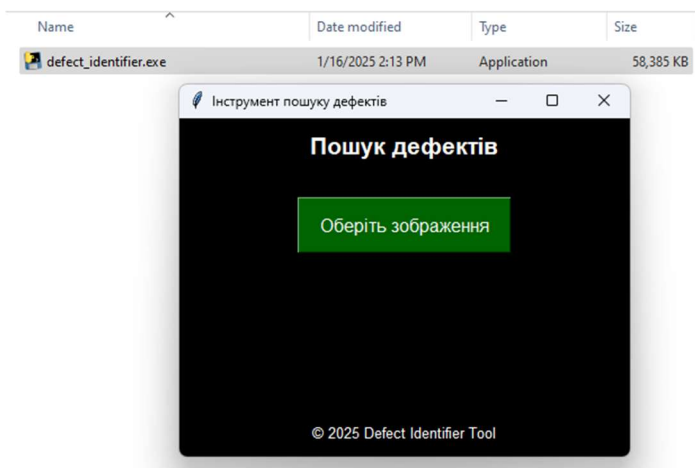


Рисунок 3.5 – Вікно програми, що відкрилось після запуску `defect_identifier.exe`

Також для тестування необхідно враховувати наявність залежностей (наприклад, Tkinter, OpenCV, тощо). Усі бібліотеки автоматично додаються в виконуваний файл.

### 3.4 Тестування програми

Для того щоб протестувати розроблений програмний засіб, необхідно виконати наступні дії:

- встановити Python, якщо він не встановлений. Перевірити наявність Python можна за допомогою запуску команди в командному рядку `python --version`. Якщо Python встановлено, ця команда покаже версію Python. Якщо ні, то необхідно його встановити: скачавши з офіційного сайту [17];
- встановити бібліотеку Tkinter, якщо ще не встановлено;
- встановити бібліотеку OpenCV виконавши команду `pip install opencv-python`. Перевірити чи успішно встановилась бібліотека, можна виконавши команду `import cv2 print(cv2.__version__)`;
- зберегти код програми в файлі з розширенням `.py`;
- редагувати файл можна в таких простих додатках як Norepad++, Sublime або якщо встановлена програма IDE на кшталт VS Code;
- програма запускається виконанням команди `python defect_identifier.py`, де «`defect_identifier.py`» назва файлу з Python програмою.

#### 3.4.1 Підготовка зображень 3D деталей для тестування

Для того щоб зменшити вплив середовища, де фотографуються надруковані деталі, необхідно очистити фон для видалення зайвих шумів. Це дозволить алгоритму сфокусуватись на обробці зображенні самих деталей замість навколишнього середовища. Є багато простих та доступних інструментів, які дозволяють працювати з фоном зображень. В даному випадку, фон було видалено та замінено на суцільний білий, що дозволить програмі працювати більш ефективно та надавати більш точні результати (рисунок 3.6).



Рисунок 3.6 – Підготовлені фото деталей без фону

### 3.4.1 Перевірка позитивного сценарію

Основним етап тестування – є перевірка чи коректно програма опрацьовує зображення деталі, яка не має дефектів. Для того щоб виконати цю перевірку, необхідно спочатку обрати еталонне зображення, а потім тестове для їх порівняння. Далі розглянемо процес перевірки успішного сценарію більш детально.

В першу чергу необхідно виконати запуск програми як на рисунку 3.7.

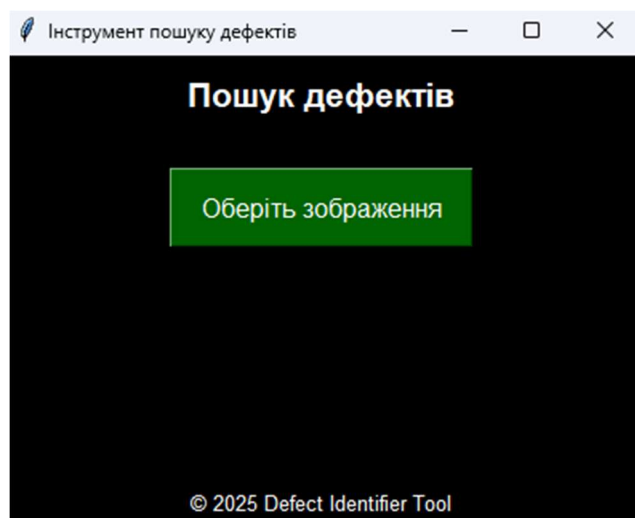


Рисунок 3.7 – Запуск основного вікна програми для тестування

Далі обираємо еталонне зображення як показано на рисунку 3.8.

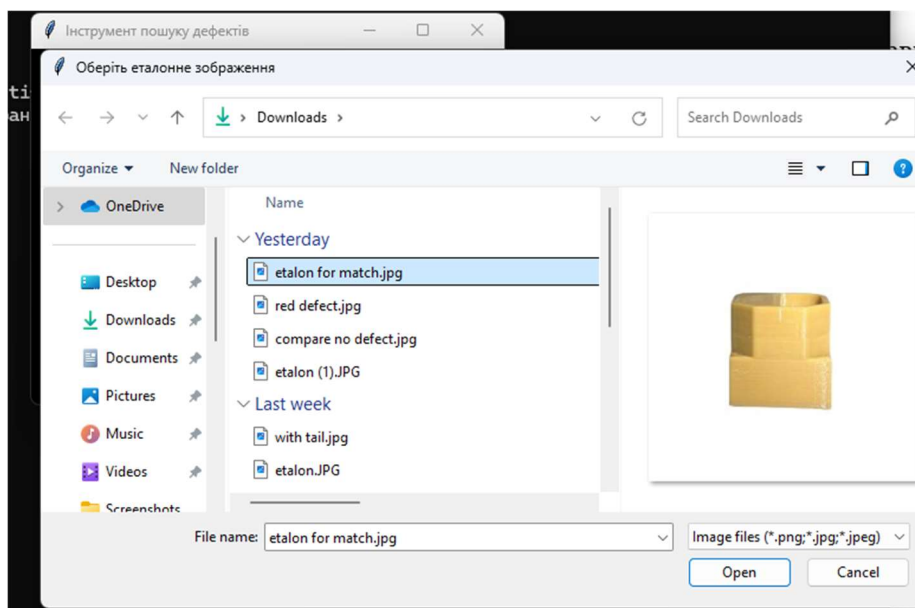


Рисунок 3.8 – Вибір еталонного зображення

Наступним кроком обираємо тестове зображення деталі, яке буде аналізуватися на предмет дефектів (рисунок 3.9).

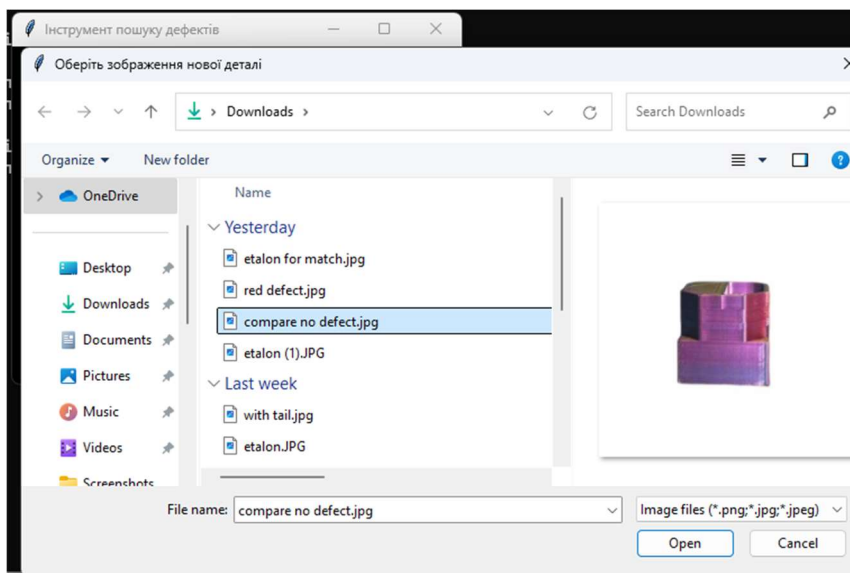


Рисунок 3.9 – Вибір тестового зображення

Далі програма виконує аналіз та показує результат – на основі виконаного алгоритму, видає результат, що дана деталь не має дефектів як на рисунку 3.10.

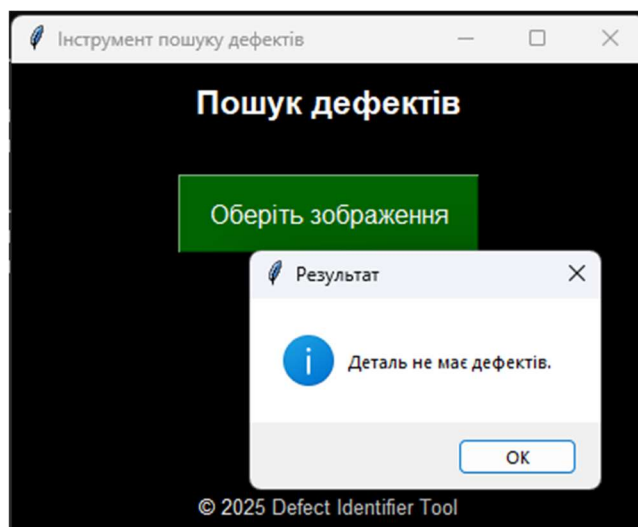


Рисунок 3.10 – Інформативне повідомлення про успішну перевірку

### 3.4.2 Перевірка негативного сценарію

Про успішну реалізацію ПЗ та його роботоспроможність можна робити висновок тільки тоді, коли вона коректно обробляє і негативний сценарій. В нашому випадку, це коли на тестовому зображенні сфотографовано деталь, яка має дефекти. Для того щоб перевірити такий сценарій, виконаємо таку послідовність дій:

Спочатку обираємо еталонне зображення як показано на рисунку 3.11.

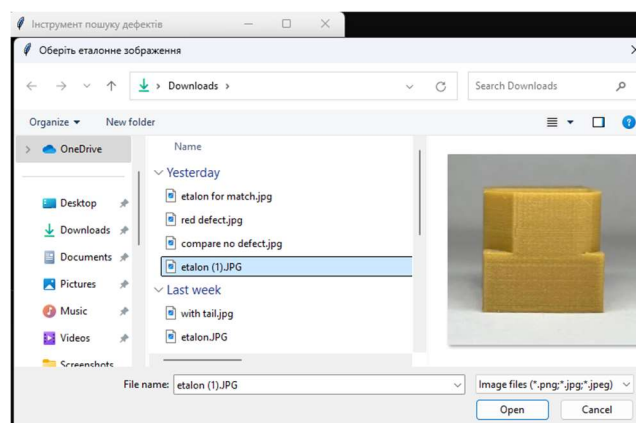


Рисунок 3.11 – Вибір еталонного зображення для тесту

1. Далі необхідно завантажити в програму фото деталі, яка має дефект як зображено на рисунку 3.12.

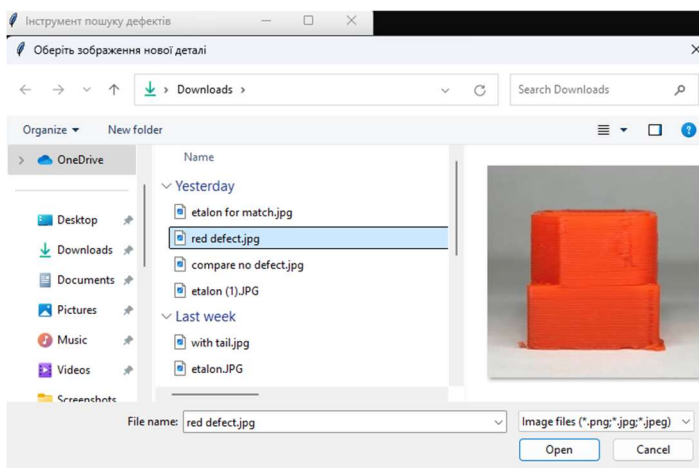


Рисунок 3.12 – Вибір зображення деталі з дефектом

На рисунку 3.13 відображено повідомлення про дефекти, що має деталь.

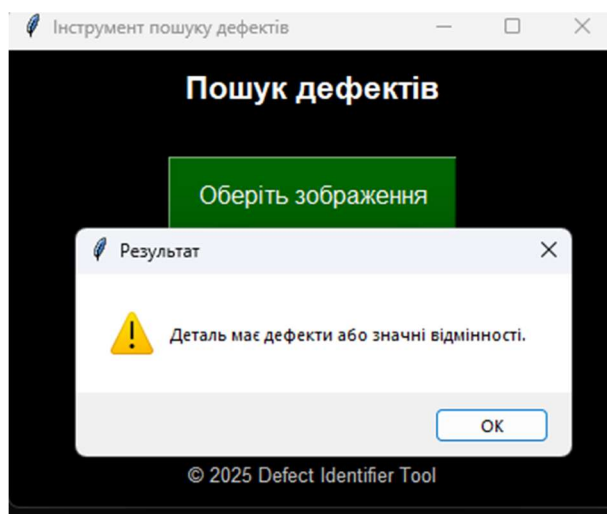


Рисунок 3.13 – Повідомлення про наявність дефектів

### 3.4.3 Перевірка вибору одного файлу двічі

Окрім роботоспроможності, якість ПЗ оцінюється не тільки за критеріями можливості виконання основної функції, а також за стійкістю до помилок і вмінням інформувати користувача про роботу програми в дружньому для користувача форматі. Наприклад, в даному застосунку, користувач може обрати одне й те саме зображення двічі, що можна помилково прийняти за позитивний результат.

Для початку вибираємо еталонне зображення (рисунок 3.14).

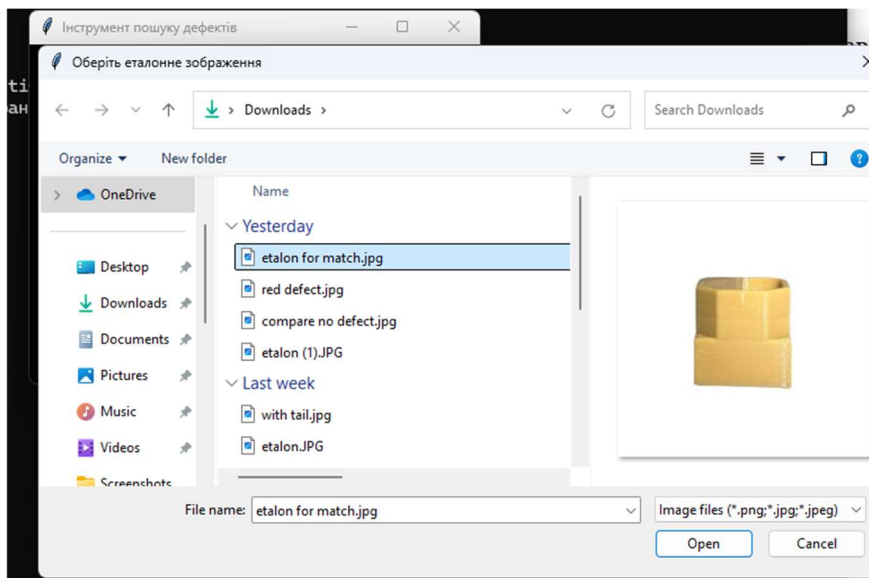


Рисунок 3.14 – Вибір зображення для тесту на дублікат

Далі обираємо те саме зображення за тим самим шляхом (рисунок 3.15).

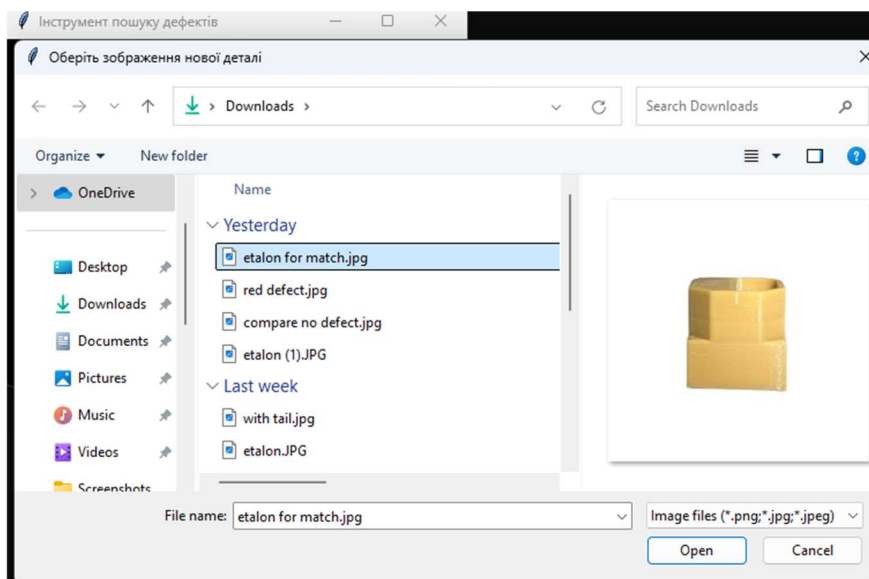


Рисунок 3.15 – Вибір другого зображення для тесту на дублікат

На рисунку 3.16 зображено як система реагує на завантаження того ж файлу.

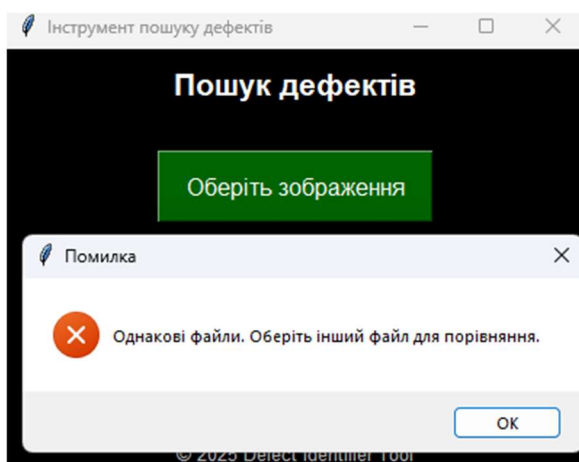


Рисунок 3.16 – Повідомлення про вибір того самого файла двічі

Для запобігання витрат часу та ресурсів на перевірку таких зображень, було додано перевірку на вибір того самого зображення за назвою, а також за вмістом. Більш детально механізм роботи такої перевірки описано в підрозділі 3.3.1.

Далі перевіряємо як програма зреагує, якщо завантажити одне й те саме зображення, але з різною назвою та з різним розташуванням. Обираємо зображення як показано на рисунку 3.17.

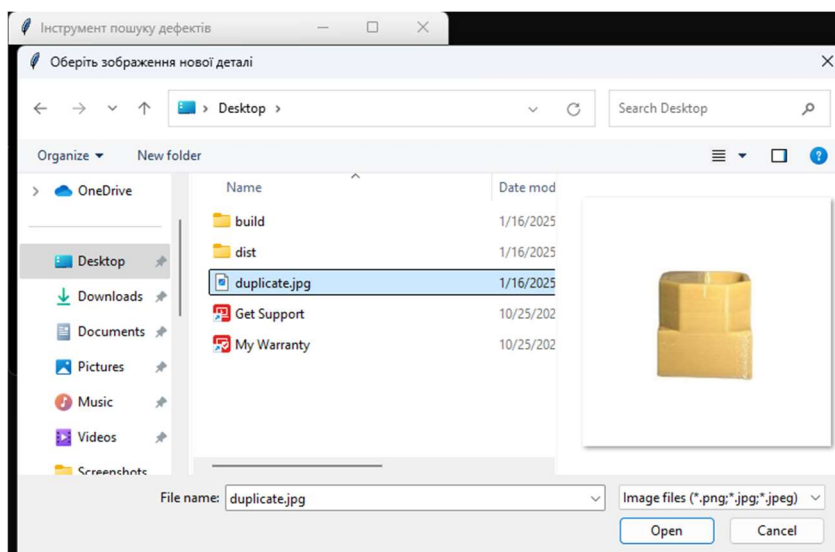


Рисунок 3.17 – Вибір другого зображення для тесту на вміст

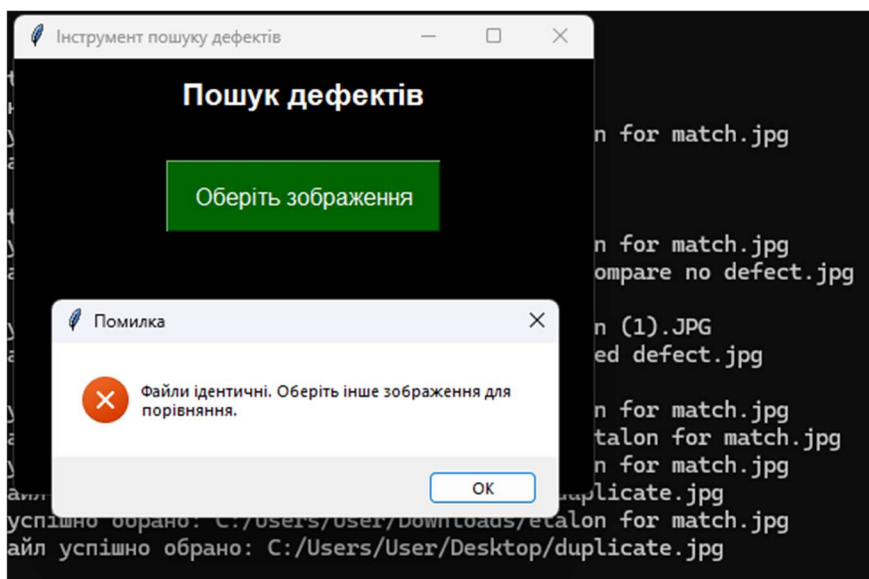


Рисунок 3.18 – Повідомлення про вибір одного зображення двічі

Як бачимо на рисунку 3.18, програма успішно обробляє та коректно реагує на обидва варіанти вибору того самого зображення. В подальшому, маючи дану перевірку в коді, програму можна буде розширювати для масової перевірки зображень з метою оптимізації часових витрат за рахунок обробки лише зображень, які відповідають вимогам перевірки. Також, в подальшому можна буде впровадити додаткові перевірки на якість зображень, що в файлі саме зображення та інші.

### 3.5 Висновки до третього розділу

У третьому розділі було розглянуто процес розробки програмного засобу для контролю якості 3D-друкованих виробів та виконано наступні етапи:

- проведено аналіз бібліотеки OpenCV, її функціональних можливостей та потенціалу для обробки зображень, зокрема у контексті пошуку дефектів у 3D-друкованих деталях;
- було здійснено порівняння мов програмування Java, C++ та Python з точки зору їхньої ефективності, зручності використання та сумісності з бібліотекою OpenCV. За результатами аналізу Python було обрано як основну мову для розробки

програмного забезпечення завдяки її простоті, великій кількості бібліотек і широкому застосуванню у сфері обробки зображень;

- створено алгоритм, який використовує методи обробки зображень для виявлення дефектів. Алгоритм базується на лінеаризації, фільтрації та згладжуванні, що забезпечує точність та надійність роботи програми;

- спроектовано інтуїтивно зрозумілий інтерфейс програми, який дозволяє користувачам легко завантажувати зображення та отримувати результати аналізу;

- визначено можливі способи розповсюдження програмного забезпечення, включаючи опенсорс-платформи та комерційне використання.

Також було виконане тестування програми, яке включало:

- перевірку позитивного сценарію, що підтвердило правильність роботи алгоритму при аналізі зображень без дефектів;

- перевірку негативного сценарію, що продемонструвало здатність програми ідентифікувати дефекти на зображеннях;

- перевірку функціоналу вибору одного файлу двічі, що підтвердило стабільність роботи інтерфейсу.

У результаті виконаних дій було створено повноцінне програмне забезпечення, яке забезпечує ефективний контроль якості 3D-друкованих виробів. Розроблений алгоритм та реалізовані функції довели свою ефективність під час тестування, що свідчить про відповідність програми поставленим завданням. Програмний засіб готовий до подальшого використання та вдосконалення.

## 4 ОХОРОНА ПРАЦІ

Робота з 3D-принтерами вимагає суворого дотримання правил охорони праці, оскільки цей процес передбачає взаємодію з нагрівальними елементами, рухомими механізмами, а також потенційно шкідливими матеріалами. Для забезпечення безпеки працівників під час використання 3D-принтерів необхідно враховувати наступні аспекти:

- щодо організації робочого місця. Перед початком роботи працівник повинен перевірити, чи правильно налаштоване та підключене обладнання, а також забезпечити відповідність робочого місця діючим стандартам безпеки. Приміщення повинно бути обладнане якісною системою вентиляції для зниження концентрації шкідливих речовин, які можуть виділятися під час друку. Робоча зона повинна бути добре освітлена для забезпечення точності виконання операцій;

- безпечно експлуатувати обладнання. Під час роботи з 3D-принтером необхідно уникати контакту з нагрівальними елементами (екструдером і платформою), температура яких може перевищувати 200–250°C. Заборонено проводити будь-які маніпуляції з принтером під час його роботи (наприклад, заміну філаменту), якщо обладнання не відключене від живлення. В разі використання УФ-випромінювання для друку або полімеризації матеріалів необхідно використовувати захисні окуляри та уникати прямого впливу світла на шкіру;

- щодо використання матеріалів. При роботі з матеріалами, які виділяють леткі сполуки (наприклад, ABS, фотополімери), обов'язково використовувати місцеві витяжки або примусову вентиляцію. Під час роботи з фотополімерними матеріалами уникати контакту з шкірою та використовувати захисні рукавички. У разі потрапляння матеріалу на шкіру або слизові оболонки – негайно промити великою кількістю води;

– вживати профілактичні заходи та реагувати на аварійні ситуації. Усі електронні компоненти принтера повинні бути захищені від вологи, щоб уникнути короткого замикання. У випадку аварійного відключення електроенергії необхідно від'єднати принтер від живлення та дочекатися відновлення стабільної роботи електромережі перед повторним запуском. Для запобігання перегріву обладнання слід регулярно перевіряти стан охолоджувальної системи;

– використовувати засоби індивідуального захисту. Такі як – захисні окуляри (для уникнення травм очей), рукавички (для захисту від хімічних матеріалів і гарячих поверхонь), респіратор (при роботі з матеріалами, що виділяють шкідливі випари);

– враховувати екологічні аспекти, а саме через відходи, що утворюються під час роботи з 3D-принтерами (невикористаний матеріал, зіпсовані деталі тощо), необхідно утилізувати відповідно до чинних екологічних норм. Використання більш екологічно безпечних матеріалів, таких як PLA або водорозчинні фотополімери, є пріоритетним.

Дотримання вищезазначених правил не лише знижує ризик травматизму та захворювань, але й сприяє ефективному та надійному виконанню робочих завдань.

## ВИСНОВКИ

У ході виконання даної кваліфікаційної роботи розроблено автоматизовану систему для контролю якості 3D-друкованих виробів. Створено ефективне програмне забезпечення для виявлення дефектів у виробах, надрукованих за допомогою FFF/FDM технології.

Виконано аналіз предметної області, а саме проведено детальне дослідження технологій 3D-друку, зокрема FFF/FDM, їхніх переваг та недоліків. Встановлено, що основними факторами впливу на якість друку є параметри налаштування принтера, властивості матеріалів і дотримання технологічного процесу. Особливу увагу приділено вивченню дефектів, таких як погана адгезія, надмірна екструзія та деформації, що є актуальними для підвищення ефективності виробництва.

Досліджено методи обробки зображень, тобто розглянуто ключові методи, такі як лінеаризація, фільтрація та згладжування, які дозволяють мінімізувати викривлення, спричинені оптичними чи фізичними факторами. Ці методи заклали основу для створення точного алгоритму виявлення дефектів у програмному забезпеченні.

Розроблено програмне забезпечення. Для цього виконано:

- вибір інструментів розробки шляхом проведеного аналізу мов програмування, сумісних із бібліотекою OpenCV, та обрано Python як оптимальний варіант завдяки його простоті використання та широкому спектру бібліотек;
- створення основного алгоритму, який використовує методи обробки зображень для виявлення дефектів;
- розробку інтуїтивно зрозумілого інтерфейсу, що дозволяє користувачам швидко та зручно завантажувати зображення й отримувати результати аналізу;
- тестування, а саме перевірено програму на позитивних та негативних сценаріях, а також на стабільність функціоналу, що підтвердило її ефективність.

Впровадження автоматизованої системи контролю якості має значний потенціал для підвищення ефективності виробництва за рахунок:

- зменшення витрат часу на ручну перевірку якості;
- підвищення точності та об'єктивності виявлення дефектів;
- скорочення виробничих витрат за рахунок зменшення браку;
- можливість масштабування системи для роботи з іншими технологіями 3D-друку та матеріалами.

Розроблена система для автоматизованого контролю якості 3D-друкованих виробів є вагомим кроком до оптимізації виробничих процесів, забезпечуючи їхню стабільність, якість і економічну ефективність.

Окрім того, що створене програмне забезпечення готове до подальшого впровадження, використання та вдосконалення для адаптації до потреб сучасного виробництва, воно також відповідає Цілям сталого розвитку України, сприяючи досягненню ЦСР 9 "Промисловість, інновації та інфраструктура" (завдання 9.2, 9.4) та ЦСР 12 "Відповідальне споживання та виробництво" (завдання 12.5).

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

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 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо-професійних програм: «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, 2023. 49 с.

3. Харківський національний університет радіоелектроніки [Електронний ресурс] /– Режим доступу: [www / URL: https://nure.ua/department/kafedra-kompyuterno-integrovanih-tehnologiy](http://www.nure.ua) avtomatizatsiyi-ta-mehatroniki-kitam/laboratorii-kafedri/studentske-konstruktorskotehnologichne-bjuro-z-robototehniki-ta-mehatroniki.

4. Найпопулярніші дефекти 3Д друку та їх усунення [Електронний ресурс] /– Режим доступу: [www / URL: https://tucans.com.ua/najpopulyarnishi-defekti-3d-druku-ta-yih-usunennya](http://www.tucans.com.ua).

5. FFF vs FDM: Difference and Best Printers [Електронний ресурс] /– Режим доступу: [www / URL: https://top3dshop.com/blog/fff-vs-fdm-difference-and-best-printers](http://www.top3dshop.com).

6. What Is FDM 3D Printing?: Revolutionizing Prototyping & Manufacturing [Електронний ресурс] /– Режим доступу: [www / URL: https://medium.com/@newayprecision/what-is-fdm-3d-printing-revolutionizing-prototyping-manufacturing-7b11ba35f638](http://www.medium.com).

7. 5 Best Free G-Code Editors for 3D Printing [Електронний ресурс] /– Режим доступу: [www / URL: https://3dprinterly.com/5-best-free-g-code-editors-for-3d-printing/](http://www.3dprinterly.com).

8. Real-time defect detection for FFF 3D printing using lightweight model deployment [Електронний ресурс] /– Режим доступу: [www](http://www) / URL: <https://link.springer.com/article/10.1007/s00170-024-14452-4>.

9. Cura Getting started [Електронний ресурс] /– Режим доступу: [www](http://www) / URL: <https://support.ultimaker.com/s/article/1667411132925>.

10. Ultimaker Cura 3D Printing Software | Complete Guide [Електронний ресурс] /– Режим доступу: [www](http://www) / URL: <https://3dism.org/ultimaker-cura-3d-printing-software-complete-guide/>.

11. Gonzalez, R. C., & Woods, R. E. Digital Image Processing. Pearson, 2018.

12. David Millán Escrivá. OpenCV 4 Computer Vision Application Programming Cookbook. Packt Publishing, 2022.

13. Getting Started with Java [Електронний ресурс] /– Режим доступу: [www](http://www) / URL: <https://dev.java/learn/getting-started/>.

14. В.В. Погребняк. Дослідження методів обробки зображень за допомогою бібліотеки OPENCV для пошуку дефектів на поверхні друкованих виробів за технологією FDM/FFF («Automation and Development of Electronic Devices» ADED-2024) [Електронний ресурс] : збірник студентських наукових статей / Харківський національний університет радіоелектроніки. Харків : ХНУРЕ. 2024. Вип. 2. с.207-213.

15. OpenCV Java Tutorials [Електронний ресурс] /– Режим доступу: [www](http://www) / URL: <https://opencv-java-tutorials.readthedocs.io/en/latest/01-installing-opencv-for-java.html/>.

16. Рейтинг мов програмування 2024. TypeScript в трійці лідерів, Python з'являється у всіх нішах [Електронний ресурс] /– Режим доступу: [www](http://www) / URL: <https://dou.ua/lenta/articles/language-rating-2024/>.

17. Python downloads [Електронний ресурс] /– Режим доступу: [www](http://www) / URL: <https://www.python.org/downloads/>.

18. OpenCV library documentation [Електронний ресурс] /– Режим доступу: [www](http://www) / URL: <https://opencv.org/>.

19. OpenCV-Python Tutorials [Електронний ресурс] /– Режим доступу: [www / URL: https://docs.opencv.org/4.x/d6/d00/tutorial\\_py\\_root.html/](http://www/URL:https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html/).

20. Технічні засоби автоматизації: підручник / І.Ш. Невлюдов, та ін. Кривий Ріг : Криворізький коледж НАУ, 2019. 366 с.

21. Alberto Fernández Villán, David Millán Escrivá. Mastering OpenCV 4 with Python. Packt Publishing, 2021.

22. E. R. Davies. Computer Vision: Principles, Algorithms, Applications, Learning. Academic press, 2021.