

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Підвищення ефективності процесу управління 3D
принтером з використанням машинного навчання»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
(код, найменування спеціальності)
освітньо-професійної програми «Інженерія програмного забезпечення»
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Іван ОЛЕЙНІКОВ
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-63

_____ Іван ОЛЕЙНІКОВ

Керівник:
д.т.н., професор

_____ Вікторія ЖЕБКА

Рецензент:
*науковий ступінь,
вчене звання*

_____ Ім'я, ПРИЗВИЩЕ

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« _____ » _____ 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Олейнікову Івану Анатолійовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Підвищення ефективності процесу управління 3D принтером з використанням машинного навчання

керівник кваліфікаційної роботи Вікторія ЖЕБКА д.т.н., професор,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» жовтня 2023р. №145.

2. Строк подання кваліфікаційної роботи «29» грудня 2023р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, методи машинного навчання, вимоги до 3D друку.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Дослідження принципів управління 3D друку.

2. Аналіз технологій машинного навчання та можливості застосування в 3D принтерах.

3. Розробка вимог підвищення ефективності управління 3D принтером.

5. Перелік графічного матеріалу: *презентація*

1. Вивчення нейромереж.
2. Процес навчання.
3. Характеристики нейромереж.
4. Результат навчання.
5. Веб-інтерфейс.

6. Дата видачі завдання «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	19.10-05.11.23	
2	Вивчення матеріалів для аналізу розвитку технологій комп'ютерного зору	06.11-12.11.23	
3	Дослідження хмарних технологій	13.11-19.11.23	
4	Аналіз особливостей ідентифікації дефектів нейромережею	20.11-26.11.23	
5	Дослідження технологій машинного навчання	27.11-03.12.23	
6	Застосування машинного навчання в роботі 3D принтеру	04.12-10.12.23	
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
8	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти

_____ (підпис)

Іван ОЛЕЙНІКОВ

Керівник
кваліфікаційної роботи

_____ (підпис)

Вікторія ЖЕБКА

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: стор. 112, табл. 5, рис. 16, джерел 37.

Мета роботи – дослідження машинного навчання комп'ютерного зору для пошуку дефектів та налаштування, розробка і застосування програмних засобів для ефективного керування 3D принтеру.

Об'єкт дослідження – вирішення задач навчання комп'ютерного зору та створення системи віддаленого керування.

Предмет дослідження – технологій машинного навчання для ідентифікації дефектів друку та програмні засоби для роботи системи керування.

Короткий зміст роботи: У роботі проведено дослідження апаратного забезпечення для функціонування системи керування, програмних засобів для роботи віддаленого керування та їх модифікація, процес навчання нейромережі. Проаналізовано основні принципи навчання нейромережі яка використовує комп'ютерний зір. Проаналізовано роботу машинного навчання для ідентифікації дефектів друку.

КЛЮЧОВІ СЛОВА: 3D ПРИНТЕР, НЕЙРОМЕРЕЖІ, МАШИННЕ НАВЧАННЯ, СИСТЕМА КЕРУВАННЯ ДРУКУ.

ABSTRACT

Text part of the master's qualification work: 112 pages, 5 pictures, 16 tables, 37 sources.

The purpose of the work is the study of machine learning of computer vision for defect detection and adjustment, development and application of software tools for effective control of a 3D printer.

Object of research – solving computer vision training problems and creating a remote control system.

Subject of research – machine learning technologies to identify print defects and software for the operation of the control system.

Summary of the work: In the work, a study of hardware for the operation of the control system, software tools for the operation of remote control and their modification, the process of learning a neural network was carried out. The basic principles of learning a neural network that uses computer vision are analyzed. The work of machine learning for the identification of print defects is analyzed.

KEYWORDS: 3D PRINTER, NEURAL NETWORKS, MACHINE LEARNING, PRINT CONTROL SYSTEM.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1 ОГЛЯД МЕТОДІВ УПРАВЛІННЯ 3D ПРИНТЕРАМИ	12
1.1 Первинний аналіз 3D принтерів екструзійного типу.....	12
1.2 Аналіз системи керування механікою.....	14
1.3 Алгоритм початку роботи пристрою друку.....	15
1.4 Аналіз компонентів друку.....	16
1.5 Програмний засіб для створення файлу інструкцій.....	18
1.6. Недоліки роботи 3D принтера.....	19
1.7. Методи управління.....	24
1.8. Рішення по автоматизації FDM принтеру.....	25
1.9 Система безперебійного живлення.....	50
РОЗДІЛ 2 ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДО УПРАВЛІННЯ 3D ПРИНТЕРОМ	55
2.1 Наукові досягнення у сфері використання нейромереж.....	55
2.2 Особливості методів машинного навчання для управління 3D принтером.....	61
2.3 Моделі штучного інтелекту для Raspberry Pi.....	70
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМИ ЗАБЕЗПЕЧЕННЯ ТА НАЛАШТУВАННЯ РОБОТИ 3D ПРИНТЕРА	78
3.1 Робота з веб-інтерфейсом.....	78

3.2	Додатковий мобільний додаток.....	81
3.3	Програмні рішення.....	85
3.4	Підготовка навчання машинного зору.....	87
3.5	Підготовка даних і розмітка даних.....	88
3.6	Навчання моделі.....	90
3.7	Дослідження параметрів процесу управління 3D принтером.....	96
3.8	Перший результат навчання нейромережі.....	101
	ВИСНОВКИ.....	104
	ПЕРЕЛІК ПОСИЛАНЬ.....	105
	ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	108

ВСТУП

3D Принтер – це розвинена та розповсюджена механічна конструкція, здатна наплавляти пластик по трьом осям. Даний апарат може бути різних розмірів, може вміститись у півметровий куб і досягати розмірів кімнати, у випадку якщо він фабричний, у більшості випадків від поміщається на робочому столі і має відносну компактність, частіше використовується для власних потреб чи прототипування.

Він використовує пластик як основний матеріал для друку, але існують різні види, наприклад, полімерні принтера, які друкують речовиною, яка застигає від впливу ультрафіолету, або більш складні і дорогі принтера, які друкують металевим порошком, який розплавляє напівпровідниковий лазер.

Головною складністю принтерів є не лише необхідні знання та навички у налаштуванні роботи, а й необхідність у обслуговуванні та контролі.

Під час друку можливі дефекти, які утворюються внаслідок неправильних налаштувань та непередбачених проблем роботи механізмів, та впливу навколишнього середовища.

Деякі дефекти можна проігнорувати, але переважна більшість робить друкований об'єкт непридатним. В такому випадку слід зупинити друк, чи спробувати виправити помилку в процесі, але це не можливо робити постійно і на відстані також. Для пошуку дефектів без участі людини можна використовувати нейромережу, здатну шукати дефекти у процесі друку, такий підхід автоматизує процес керування.

Також принтер сам не здатен зняти надруковану деталь з нагріваючої платформи, це робить обслуговуючий персонал, що створює деякі складності як для звичайних користувачів так і для великих виробників які бажають спростити виробництво. Проблему здатен вирішити робот-маніпулятор з віддаленим керуванням, який зніме модель з платформи, і принтер зможе почати нову роботу.

Нейромережу, відеоспостереження та робот маніпулятор можна об'єднати в один відкритий проект, який буде здатний працювати на простому міні-

комп'ютері, що зробить процес підвищення ефективності керування принтером доступним.

Якщо поєднати сторонній комп'ютерний пристрій з платою керування принтером, то в такому випадку операційна система на комп'ютерному пристрої отримає пряму можливість керувати функціями принтеру, доступ до яких має користувач. Підключення до бездротової мережі та налаштування веб-інтерфейсу зробить принтер керованим з будь якої точки планети, що максимально збільшить ефективність керування, даючи змогу керувати принтером віддалено, додатково спростивши задачу наглядом комп'ютерного зору, та інтегруванням маніпулятора, для неперервної роботи.

Мета - дослідження та оптимізація процесів екструзійних 3D принтерів з використанням комп'ютерного зору та веб-інтерфейсу для підвищення ефективності контролю та покращення якості друку.

Завдання - розробка алгоритмів комп'ютерного зору для виявлення дефектів друку, також розробка додаткового методу керування у контексті роботу-маніпулятора, створення веб-інтерфейсу для віддаленого керування і поліпшення керування та контролю.

Об'єкт - екструзійні 3D принтери, методи керування через веб-інтерфейс і програмні засоби для його роботи, нейромережі здатні використовувати комп'ютерний зір.

Предмет - оптимізація роботи 3D принтерів за допомогою комп'ютерного зору та веб-інтерфейсу з використанням додаткового обладнання та головного комп'ютеру.

1 ОГЛЯД МЕТОДІВ УПРАВЛІННЯ 3D ПРИНТЕРАМИ

1.1 Первинний аналіз 3D принтерів екструзійного типу

FDM - Fused Deposition Modeling, що означає моделювання зі злиттям.

Це технологія 3D друку, яка використовує термопластичний матеріал, який плавиться і виштовхується через сопло на нагрівальну платформу, де він шар за шаром утворює об'єкт за цифровою 3D моделлю. Ця технологія також називається FFF (Fused Filament Fabrication), що означає виготовлення зі злиттям нитки.[1]

Також принтери на даній технології називають – екструзійними.

Сам принтер складається умовно з 8 частин розбитих на деталі:

- корпус;
- блок живлення;
- контролер;
- друкуюча голова;
- екструдер;
- направляючі деталі;
- монітор;
- електродвигуни.

Перша частина, основа, це – корпус-рама.

Як правило, виготовлений з алюмінію та пластикових елементів каркас, який також направляюча і захист для внутрішнього обладнання.

У корпусі сховані такі деталі як блок живлення та контролер.

Сам корпус може слугувати як направляюча основа для осі вертикального керування друком, наприклад “П” подібних чи “Г” подібні стійки для руху друкуючої голови по двом чи одній осі. В такому випадку корпус стає невід’ємною частиною механіки.

Також він повинен мати велику вагу, наприклад, 8 кг для стійкості під час переміщень друкуючої голови та печатної платформи, які здатні створювати

інерційні рухи під час яких можливі непередбачені зміщення, внаслідок яких можливі спотворення якості друкованого виробу.

Присутні на ринку зразки, в яких корпус надрукований на іншому принтері, в такому випадку деталі мають більші габарити, але легші, тому прикріплені до столу. Принтер, в якому велика кількість пластикових друкованих деталей, частіше виготовлений користувачами принтеру, має великий потенціал для переробки та модифікації. Фірми виробники обирають металеві деталі для масовості виробництва.

Для аналізу і подальшого підвищення ефективності контролю друку буде використано поширений тип принтеру із дешевого цінового сегменту - “Anycubic Kobra Neo”.

Принтер китайського походження має перевірене компонування, і широку аудиторію, використовує стару конструкцію, має власний дизайн та нововведення.



Рис.1.1. Anycubic Kobra Neo - екструзійний принтер з головою типу Direct.

Принтер побудований на кінематиці “Prusa”.

Prusa (пруша) – кінематика відома як “дригостол” рухає ліжко по осі Y, в той же момент голова рухається окремо по осі X. Така проста кінематика найрозповсюдженіша і полегшує роботу для подальших модифікацій.

Наприклад, якщо додати маніпулятор, то друкуючу голову можна підняти максимально вгору, а ліжко перемістити вперед, тоді зняти з поверхні друку модель і маніпулювати спеціальним роботом буде просто.

Більша частина корпусу виготовлена з алюмінієвих профілів спеціального призначення, деякі деталі пластикові, корпус для комп'ютеру схований у пластиковий захист, присутнє активне охолодження. Конструкція виготовлена якісно та надійно.

1.2. Аналіз системи керування механікою

Механіка пристрою має власний керуючий комп'ютер та програму. Програмний код встановлений у контролер принтеру відомий під назвою – “прошивка”, вона ж примітивна операційна система для сприйняття і обробки файлу інструкцій, також надає можливість втручання у роботу принтеру. Найпопулярніша прошивка відкритого коду – “Marlin”.

Marlin - це відкрита прошивка для 3D-принтерів, яка керує рухом друкуючої голови та іншими аспектами друку. Вона є однією з найпопулярніших та широко використовуваних прошивок у 3D-друку.

Особливості:

- **Відкритий код:** Marlin розповсюджується на умовах ліцензії GPL (General Public License), що робить її відкритою та доступною для змін та розповсюдження. Сприяє розвитку спільноти та обміну ідеями.
- **Керування Рухом:** Marlin відповідає за керуванням руху друкарської головки по трьом основним координатам (X, Y, Z). Вона взаємодіє з кроковими моторами, визначаючи їхні кроки та напрямки для точного переміщення.
- **Екструдер і Матеріали:** Прошивка керує роботою екструдера, відповідального за подавання та екструзію пластикового матеріалу. Marlin дозволяє конфігурувати налаштування для роботи з різними матеріалами та типами екструдерів.
- **Авторівень Ліжка:** Одна з корисних функцій Marlin - це можливість використовувати автоматичне вирівнювання руху голови вздовж ліжка відносно нерівностей та різного рівня чотирьох країв поверхні, що дає

можливість не регулювати положення вручну. Полегшує процес налаштування перед кожним друком.

- **Графічний Інтерфейс та Керування:** Marlin підтримує LCD-екрани та інші способи взаємодії, що дозволяє користувачам змінювати налаштування та керувати принтером безпосередньо з пристроєм.
- **Термальний Контроль:** Прошивка забезпечує термальний контроль різних елементів принтера, таких як гарячий екструдер, ліжка, та інші. Це важливо для запобігання перегріву та забезпечення безпеки використання.
- **Розширені Функції:** Marlin підтримує багато різноманітних функцій, таких як підтримка множинних екструдерів, відстеження наявності філаменту, пристосування швів, інтерполяція шляху та інші, що полегшують роботу з принтером та покращують якість друку.

Marlin є гнучкою та налаштованою прошивкою, яка може бути адаптована до різних моделей 3D-принтерів та вимог користувачів. Спільнота користувачів та розробників Marlin активно працює над її постійним вдосконаленням і розширенням функціоналу.

Прошивка не викликає труднощів та не є необхідною до модифікування.

1.3. Алгоритм початку роботи пристроєм друку

Перед тим як почати роботу, пристрій запускається кнопкою живлення, після чого відразу включається блок живлення та його постійно працюючий кулер охолодження, який викликає незручності через шум.

Після включення блоку живлення, запускається система принтеру, робота якої представлена на моніторі-енкодеру. Якщо монітор не працює, це системна помилка, можливо система перевантажена і необхідно перезапустити апарат або витягнути карту пам'яті.

Після запуску файлу друку починається робота нагрівальних елементів.

Нагрівання друкуючої поверхні → нагрівання “Хот-енду” → повернення рухомих частин у стартове положення → початок друку.

Після чого друкуюча голова почне видавлювати пластик на платформу створюючи малюнок першого слою моделі.

Слід зазначити важливі можливості запуску друку.

Перший спосіб – це встановлення спеціального файлу інструкцій для системи принтеру по якій буде йти процес. На Micro-SD карту встановлюється файл з комп’ютеру, після цього карта вставляється в спеціальний слот. Робота буде йти постійно і без затримок, однак контроль за друком з персонального комп’ютеру буде менший.

Другий спосіб – друк через USB, в такому випадку карта пам’яті не потрібна, виконуючий файл інструкцій скомпільований на комп’ютері постійно передається до системи принтеру, збільшуючи можливість керування друком, але в такому випадку не можна відключати кабель, щоб не втратити контроль та прогрес.

Висновком стає розуміння надійності першого способу, який використовує обчислювальні потужності принтеру, а не комп’ютеру.

1.4. Аналіз компонентів друку

Хот-енд – “**HotEnd**”(хот-енд) – це назва для нагрівального елемента, в який встановлене сопло, в ньому пластик нагрівається та виходить через сопло певного діаметру, наприклад від 0.2 до 0.8.

Слід зазначити що сопла є різних видів і на кожний різний діаметр. Наприклад, є розповсюджені сопла типу “V8”, їх діаметр від 0.2 до 0.8 мм дає змогу обирати між якістю та швидкодією.

При виборі сопла меншого діаметру, можна друкувати тонші стінки які будуть утворювати гостріші краї шару, вони будуть менш округленими і якісними.

Більший діаметр дасть змогу видавлювати більше пластику за той же час роблячи стінки круглішими, а шари моделі більш помітними.

Насправді ширина шару не залежить від ширини сопла, програмно можна збільшити ширину доріжки і таким чином екструдер стане подавати більше чи менше матеріалу в той час коли програма робить ширші відстані між доріжками.

Екструдер – набір з електродвигуна, опори та шестерні подачі, він відповідає за подачу пластику в хот-енд.

Шаговий електродвигун з передачею, яким керує комп'ютер принтеру, задаючи необхідну кількість кроків на момент часу для регулювання швидкості подачі пластику.

Відомі два типи: **Bowden** та **Direct**.

Боуден (Bowden) – найбільш розповсюджений і простий спосіб компонування, в такому випадку екструдер знаходиться на рамі і філамент подається через тefлонову трубку у хот-енд, трубка довжиною у 30 см і ширшим за пластик діаметром внутрішніх стінок..

Плюси:

- Швидкість друкуючої голови.
- Легкість обслуговування.

Мінуси:

- Складно друкувати м'якими пластиками.
- Труднощі з трубкою.

Direct – екструдер який не має таких проблем за рахунок знаходження екструдера у друкуючій голові, в такому випадку вона важка і повільніша, але вже немає минулих проблем, зручний варіант і універсальний.

Плюси:

- Не треба обслуговувати часто.
- Легше налаштувати друк м'якими матеріалами.

Мінуси:

- Повільна.
- Велика інертність рухів.

Шлях від механізму подачі мотору до хот-енду дуже короткий, тому ним набагато легше друкувати гнучким матеріалом, який може розтягуватись і псувати друк, утворюючи наплив пластику.

Саме такий тип на даному принтері, однак він також має можливість утворення пробки з розплавленого пластику на шляху до хот-енду при неправильній експлуатації, тут постає складність з обслуговуванням, ліквідація проблеми потребує розборки друкуючої голови.

Механіка працює за рахунок електродвигунів. Електромотори у принтері відіграють ключову роль у його функціонуванні.

Використовуються крокові мотори. Це найпоширеніший тип моторів у принтерах. Вони дозволяють точно контролювати довжину руху, що є важливим для точного переміщення друкарської головки та подачі пластику.

Функції електромоторів у принтері – це переміщення друкарської голови, переміщують голову принтера вздовж осі.

Електромотори керуються за допомогою електроніки принтера, яка отримує інструкції від карти пам'яті або від комп'ютера USB-порт.

Крокові мотори керуються імпульсами, що дозволяє точно визначати їх положення.

1.5. Програмний засіб для створення файлу інструкцій

Робота принтера можлива завдяки програмам типу “Слайсер”.

Слайсер (Slicer) – вид спеціалізованих програм, розділяє модель на шари і по внутрішньому діаметру утворює маршрут, шлях для сопла, задаючи такі параметри як обдув, і подача пластику а також швидкість та багато інших.

Cura – популярний і універсальний варіант, який підійде до будь якого принтеру.

Здатна створити файл “**gcode**” який можна відправити у внутрішню пам'ять принтеру як керуючий друком файл, або виконавши підготовку моделі до друку використати функцію друку через USB. Другий варіант дає можливість

використовувати саме цю програму для принтерів, які приймають інший формат керуючого файлу, але друк повинен бути при постійному підключенні між принтером і комп'ютером, на якому виконується програма.

Складність для підвищення ефективності контролю друку та для автоматизації полягає у двох особливостях програми та міні-ПК на процесорі ARM, Cura не має версію для роботи на системах під керуванням цих процесорів, тому слід звернути увагу на інші варіанти:

- Використання емулятору - таких як QEMU, для запуску на Raspaberry.
- Використання Windows для ARM систем.
- Використання іншого програмного засобу - PrusaSlicer или AstroPrint.

Принтера працюють завдяки файлу загального формату "Gcode".

Gcode є файлом, який зберігає набір команд та інструкцій для станків типу ЧПУ. Наприклад код "M140" визначає включення функції автовирівнювання, цей код один з початкових, тому друкуюча голова під час друку буде корегувати своє положення відносно стола по осі Z, опускаючись там, де стіл нижчий відносно рівня голови, корисна і обов'язкова функція якщо стіл не регульований і не рівний.

В самому кінці коду знаходяться кінцеві коди деактивації підігрівів, активності охолодження та повернення столу та голови у стандартне положення.

1.6. Недоліки роботи 3D принтера

Головним недоліком процесу друку є необхідність в контролі за роботою, через вірогідність помилок і дефектів друку.

Найбільш частою проблемою є неправильне налаштування параметрів друку, що по більшій мірі вирішується завдяки знанням та навичкам.

Однак, бувають проблеми, які здатні зіпсувати будь-яку модель, особливо, коли друк йде більше доби.

Головні проблеми:

- недоекструзія,
- помилка висоти першого слою,
- переекструзія,
- перегрів,
- недогрів,
- забиття сопла і механізму подачі,
- критичні помилки програми принтеру,
- помилка налаштування швидкості друку,
- падіння підтримки,
- помилки відновлення роботи після паузи чи втрати живлення,
- обрив філаменту.

Всі перераховані проблеми мають чіткі причини і давно знайомі користувачам.

Таблиця 1.1

Недоліки роботи 3D принтера

№	Назва	Опис	Причини
1	Недостатня екструзія	Недостатня подача матеріалу, виражена у пустотах між шарами по горизонталі, а також у пустотах між рівнями шарів, нерівні стінки і слабка конструкція.	Малий діаметр філаменту, недостатня температура, слабкий прижим механізму подачі, засмічення сопла, пробка, помилка налаштування.
2	Надмірна екструзія	Завелика подача матеріалу, виражається як локальне явище коли було подано забагато пластику і утворено наплив. Також, можливо, що шари виходять за рамки моделі постійно або на конкретних шарах, також випадково.	Перегрів, помилка налаштування, пробиття пробки, завеликий діаметр філаменту, несправність.
3	Павутиння	Подача матеріалу на відсутню поверхню, після того як відпала підтримка чи частина моделі відклеїлась, також, коли підтримка не була передбачена у налаштуваннях проекту, виглядає як випадкові волоси чи спагеті.	Відсутність підтримки, падіння підтримки, помилка висоти між шарами, відрив частини моделі.

Продовження таблиці 1.1

Недоліки роботи 3D принтера

№	Назва	Опис	Причини
4	Викривлення	загинання країв моделі, задрані краї псуують якість стінок, можуть призвести до завеликого тиску на модель і вона відлипне. У рідких випадках попередній шар буде нижче і верхній не зможе наплавитись утворивши пустоти.	Слабкий обдув, велика швидкість друку, відсутність підтримки при сильно нависаючому куті.
5	Зміщення слоїв	невідповідність положення слою відносно по вертикалі чи горизонталі. Локальна чи постійна невідповідність положення шарів по горизонталі, робить друкування точних деталей неможливим і призводить до непередбачуваних результатів.	Хибне регулювання прижиму підшипників, слабе чи сильне затягування ременя передачі, помилка роботи контролера, помилка кодування виконуючого файлу, несправність.
6	Вібрації слоїв	Нерівномірність стінок, вібрації на поверхні. Стінка втрачає пряму текстуру, стає нестабільною у плані рівності і гладкості, шумове переливання світла.	Велика вібрації від рухів кінематики завелика швидкість, перетягування ременів передачі руху шагового мотору, втрата ресурсу двигунів.
7	Пухирі та перегрів	Шароподібні надлишки матеріалу, спінення, вихід за передбачені межі шарів, нерівний блиск від поверхні, або його втрата.	Вологий пластик, затримки на шляху шару, помилка налаштування, завелика температура, особливості матеріалу.
8	Помилка першого слою	Положення сопла над столом, високе чи низьке, призводить до слабкої адгезії і ненадійності друку або перші шари страждають від надлишку матеріалу.	Неправильне налаштування проміжку між столом та соплом призводить до раннього відлипання моделі.

Недоекструзія може бути через помилку налаштувань у програмі слайсеру, де задається процентне співвідношення подачі пластику до голови, через великий відкат пластику для того щоб пластик не витікав з сопла під час переміщення, але

великий відкат може утворити пустоти у камері нагріву, неправильний діаметр філаменту, все це також в оберненій перспективі стосується переекструзії.

Перший шар у кожному принтері повинен бути налаштований заздалегідь. У програмі не можна налаштувати положення сопла відносно столу, тільки висоту першого слоя, однак можна змінити налаштування файлу друку так, щоб сопло змінило висоту перед друком.

Наприклад якщо висота першого слоя 0.2 мм використовується постійно, то треба у налаштуваннях самого принтеру вручну підібрати висоту між соплом та столом, бо після фабрики зазор може сягати декілька міліметрів, що робить друк неможливим.

Перегрів часта проблема новачків через незнання температурного режиму.

Під час перегріву пластик втрачає свої властивості, піниться та утворює запах.

Щоб запобігти цій проблемі треба знати температурні рамки матеріалу, а також не порушувати співвідношення температури та швидкості подачі яка залежить від друку, пластик не повинен довго знаходитись у соплі бо він неменуче під дією температури спіниться, тому важливо не друкувати у надто повільному темпі.

Недогрів пластику буде видно візуально, наприклад, деякі криві лінії стінок не встигнуть припаятися та зіжмуться ставши прямими у неправильному місці, наприклад, замість кривої пластик стягнеться у пряму від охолодження. Недостатня екструзія теж як наслідок буває, там де велике нависання і мала площа для стінки шару, або відсутня зовсім, тоді необхідна підтримка.

Складно передбачити, коли сопло заб'ється, бувають дефекти друку, які говорять про проблему у подачі пластику, наприклад, недоекструзія.

Також може поплавитись тефлонова трубка на шляху до хот-енду, і сплавитись з пластиком і пластик видавить туди де температура недостатня для плавлення, і потрібне перебирання компонентів голови.

На жаль, трапляються випадки критичних помилок, трапляються дуже рідко, але не запобігти, однак можна перезапустити процес з моменту помилки, але в

такому випадку трапляються незбіжності з положенням сопла чи інші непередбачувані проблеми, тут треба ручне налаштування.

Трапляється помилка пов'язана з властивостями матеріалу, бувають матеріали які при температурі плавлення сильно густі і довго видавлюються з сопла, і не встигають лягти на стінку, і друкувати треба повільно, або матеріал має велику усадку і при зашвидкому друку краї нависань задираються і псують кінцевий вигляд, чи сжимання на цьому місці і товщини слою не вистачає для злипання.

Підтримки - важливі структури, які генерує програма слайсер. Там де нависання має великий процент потрібна підтримка, щоб доріжка мала основу по якій може налипати, бо принтер не здатен друкувати до повітря. Самі ж підтримки як правило однослойні, з прямих стінок, зігзагом чи трубами, які називають дерева, і основа в них теж однослойна. Якщо перегріти стіл, то основа стане гнучкою і коли підтримка стане надто високою, то під впливом руху сопла, вона втратить стабільність, зігнеться або відірветься.

Також філамент може закінчитися, і часто в принтерах є датчик кінця філаменту. Конкретно в наявному принтері його немає, і якщо філамент закінчиться, то він буде друкувати без пластику, роблячи пусті рухи.

Інші проблеми вже не такі часті, як бульбашки від пауз під час друку, бо раніше процесор принтеру міг не встигати обробляти код і робив паузу, під час якої витікав пластик.

1.7. Методи управління

Ручне управління.

Користувач керує принтером за допомогою енкодера, екрану з кнопочним керуванням, або за допомогою сенсорного екрану.

На цих екранах є дані температур столу та голови, сила роботи обдуву, та прискорення у процентах від 10 відсотків від швидкості заданої у файлі до 300

процентів, також відстань у сотих міліметра між столом і датчиком столу друкуючої голови, так задається проміжок для вдалого першого слою.

Також є меню в якому можна включити такі функції як:

- друк з SD карти,
- підготовка нагріву для певного виду пластику,
- подача нового чи витягання старого філаменту,
- замір нерівності столу,
- ручне керування положенням осей,
- модифікувати потік у процентах,
- змінити швидкість,
- поставити паузу чи відмінити друк,
- відновити друк,
- мова,
- конфігурація.

Ці функції дають змогу виправити процес друку, навіть якщо швидкість зависока, малий обдув, мала подача та інше. Тобто є можливість коригувати процес під час друку, що економить час і зменшує кількість браку.

Мануальний спосіб – коли користувач сам слідкує за роботою.

Можливий і інший спосіб, який використовує комп'ютер.

Одним із таких може бути одноплатний комп'ютер “Raspberry Pi”, він здатен при правильному підключенні отримувати дані від принтеру напряму, сам буде зберігати і виводити дані з енкодеру дублюючи їх.

В такому випадку за допомогою кодування можна створити програмний засіб з інтерфейсом, який дублює і розширює дані та функціонал. Є можливість розробити програму для комп'ютеру чи телефону, яка дасть змогу керувати принтером віддалено.

Ситуація, в якій ви у від'їзді, а принтер працює, трапляється часто, але при підключенні камери до raspberі утворюється можливість слідкувати за роботою у реальному часі як мінімум з однієї камери, бо на платі комп'ютеру є модуль Wi-Fi.

Перший крок до автоматизації, створити умови для нагляду і керування роботою принтеру, це фундамент для подальшої роботи.

Наступним кроком стане розробка маніпулятора, для дистанційного зняття надрукованої моделі з друкарського столу.

Завдяки камері можна буде на відстані керувати маніпулятором вручну через програму.

Сам маніпулятор має декілька моторів, і мати повинен змінні насадки, одна з лезом для піддівання моделі і відділення, інша у вигляді клешні, щоб взяти модель чи замінити філамент при його закінченні чи обриві.

Після цього починається розробка програмного засобу на основі машинного навчання, де комп'ютерний зір здатен відрізнити недоліки роботи і дефекти, і нейро технологія зробить логічний висновок та застосує корегування чи почне заново друк.

Також присутня черга для друку дасть змогу задавати друк на дні наперед без необхідності втручання, створюючи міні завод автономного керування і часткового обслуговування.

1.8. Рішення по автоматизації FDM принтеру

Першочергово продвинуті користувачі принтерів встановлюють одноплатний комп'ютер, який здатен приймати дані від принтеру і керувати його функціями.

Маючи модуль Wi-fi такий комп'ютер здатен відправляти та приймати дані стаючи інтерфейсом віддаленого керування завдяки наявності спеціальних програм для телефону чи персонального комп'ютера. Існують у різних версіях та збірках, наприклад не завжди потрібна пам'ять на 8 Гб, тому купують дешевшу версію з 2 Гб, різних фірм та виробників.

Наприклад, зараз доволі відомі і популярні міні-комп'ютера "Raspaberry Pi".

Raspberry Pi - це міні-комп'ютер, який можна використовувати для багатьох завдань, включаючи керування 3D-принтером.

Raspberry Pi розроблений у Великій Британії фондом Raspberry Pi.

З моменту випуску першої версії Raspberry Pi у квітні 2012 року було випущено 15 різних версій:

1. **Raspberry Pi 1 Model B** (2012): перша версія Raspberry Pi, яка була випущена у квітні 2012 року. Вона мала одноядерний процесор ARM11 із тактовою частотою 700 МГц, 256 МБ оперативної пам'яті, Ethernet-порт, 2 порти USB, HDMI-порт та слот для карт пам'яті SD.

2. **Raspberry Pi 1 Model A** (2013): це була урізана версія Raspberry Pi 1 Model B, випущена в лютому 2013 року. Вона мала ті самі характеристики, що і Model B, але без Ethernet-порту та одним USB-портом.

3. **Raspberry Pi 1 Model B+** (2014): це була покращена версія Raspberry Pi 1 Model B, яка була випущена у липні 2014 року. Вона мала самі характеристики, як і Model B, але з 512 МБ оперативної пам'яті, 4 портами USB і слотом для карт пам'яті MicroSD.

4. **Raspberry Pi 2 Model B** (2015): це була перша версія Raspberry Pi, яка використовувала чотириядерний процесор ARM Cortex-A7 із тактовою частотою 900 МГц та 1 ГБ оперативної пам'яті. Вона також мала 4 порти USB, Ethernet-порт, HDMI-порт та слот для карт пам'яті MicroSD.

5. **Raspberry Pi Zero** (2015): це була найкомпактніша і найдоступніша версія Raspberry Pi, яка була випущена в листопаді 2015 року. Вона мала одноядерний процесор ARM11 із тактовою частотою 1 ГГц, 512 МБ оперативної пам'яті, слот для карток пам'яті MicroSD та міні-HDMI-порт.

6. **Raspberry Pi 3 Model B** (2016): це була перша версія Raspberry Pi, яка використовувала чотириядерний процесор ARM Cortex-A53 із тактовою частотою 1,2 ГГц та 1 ГБ оперативної пам'яті. Вона також мала 4 порти USB, Ethernet-порт, HDMI-порт та слот для карт пам'яті MicroSD.

7. **Raspberry Pi Zero W** (2017): це була покращена версія Raspberry Pi Zero, яка була випущена у лютому 2017 року. Вона мала ті ж характеристики, що і Raspberry Pi Zero, але за допомогою Wi-Fi і Bluetooth.

8. **Raspberry Pi 3 Model B+** (2018): це була покращена версія Raspberry Pi 3 Model B, яка була випущена у березні 2018 року. Вона мала ті ж характеристики, що і Model B, але з чотириядерним процесором ARM Cortex-A53 із тактовою частотою 1,4 ГГц, 1 ГБ оперативної пам'яті, підтримкою Wi-Fi та Bluetooth.

9. **Raspberry Pi 3 Model A+** (2018): це була урізана версія Raspberry Pi 3 Model B+, випущена в листопаді 2018 року. Вона мала ті самі характеристики, що і Model B+, але без одного порту USB, меншої оперативної пам'яті (512 МБ) та меншої ціни.

10. **Raspberry Pi 3 Model B+** (2018): це була покращена версія Raspberry Pi 3 Model B, яка була випущена у березні 2018 року. Вона мала ті ж характеристики, що і Model B, але з чотириядерним процесором ARM Cortex-A53 із тактовою частотою 1,4 ГГц, 1 ГБ оперативної пам'яті, підтримкою Wi-Fi та Bluetooth.

11. **Raspberry Pi 4 Model B** (2019): це була перша версія Raspberry Pi, яка використовувала процесор ARM Cortex-A72 з тактовою частотою 1,5 ГГц та 1, 2, 4 або 8 ГБ оперативної пам'яті. Вона також мала 2 порти USB 2.0 та 2 порти USB 3.0, Ethernet-порт, 2 HDMI-порти та слот для карт пам'яті MicroSD.

12. **Raspberry Pi 400** (2020): це була унікальна версія Raspberry Pi, випущена в листопаді 2020 року. Вона є комп'ютером в корпусі клавіатури, який містить процесор ARM Cortex-A72 з тактовою частотою 1,8 ГГц, 4 ГБ оперативної пам'яті, 2 порти USB 3.0, Ethernet-порт, 2 HDMI-порти і слот для карт пам'яті MicroSD.

13. **Raspberry Pi Pico** (2021): це була перша версія Raspberry Pi, яка не є повноцінним комп'ютером, а скоріше мікроконтролером. Вона містить процесор ARM Cortex-M0+ з тактовою частотою 133 МГц, 264 КБ оперативної пам'яті, 2 порти GPIO та слот для карток пам'яті Micro-USB.

14. **Raspberry Pi Compute Module 4** (2020): це була версія Raspberry Pi, розроблена для використання в промислових додатках. Вона містить

процесор ARM Cortex-A72 з тактовою частотою 1,5 ГГц, 1, 2, 4 або 8 ГБ оперативної пам'яті, 32 ГБ eMMC-пам'яті, 2 порти USB 2.0 і 2 порти USB 3.0, Ethernet-порт, 2 HDMI-порти та слот для карт пам'яті MicroSD.

15. **Raspberry Pi 5 Model B (2023)**: це остання версія Raspberry Pi, випущена в 2023 році. Вона містить процесор ARM Cortex-A78 з тактовою частотою 2,5 ГГц, 8 ГБ оперативної пам'яті, 64 ГБ eMMC-пам'яті, 2 порти USB 2.0 та 2 порти USB 3.0, Ethernet-порт, 2 HDMI-порти та слот для карт пам'яті MicroSD

З них лише декілька мали Wi-Fi модуль без якого користування в тематиці магістерської роботи було б не зручним, звісно більшість має порт Ethernet, але слід уникати зайвих проводів.

Ті які мають модуль радіо зв'язку:

- Raspberry Pi Zero W;
- Raspberry Pi 3 Model B;
- Raspberry Pi 3 Model B+;
- Raspberry Pi 4 Model B;
- Raspberry Pi 400;
- Raspberry Pi 5 Model B.

З них слід обирати який стане основою для подальшої автоматизації принтеру. Слід зазначити, що основною причиною вибору може стати ціна, а вже потім потужність.

Перший варіант – Raspberry Pi 3 B.

Причина: версія під номером 3 вже на ринку декілька років, було випущено багато екземплярів, тільки у 2018 році Pi 3 Model B було випущено більше 9 мільйонів [3], наразі число може бути більше ніж десятки мільйонів, також випуск нових версій розбавив ринок і зменшив цінність моделей під номером 3.

Слід обрати “**Raspaberry Pi 3 Model B+**” як початковий варіант, дешевий, система повинна першочергово бути доступною, а завдяки їх подібності заміна на новішу модель не призведе до складнощів несумісності, так як новіші моделі просто потужніші і швидші в них немає критичних та значущих змін.

Другий варіант – Raspberry Pi 4 B.

Причина: має більшу потужність, завдяки чому робота операційної системи та машинного зору стане надійною та швидкою.

Підключення даних комп'ютерів можливе через USB-порт.

Другий основний елемент – Камера.

Існує безліч камер, які можуть використовуватися з Raspberry Pi 3, для списку задач який починається з спостереження. В такому випадку через модуль Wi-Fi будуть відправлятися дані на телефон чи комп'ютер і можна віддалено спостерігати за процесом друку.

В такому випадку при умові підключеного до принтеру міні-комп'ютеру, буде можливість під час спостереження в режимі онлайн регулювати роботу принтеру на відстані, а саме - змінити температуру сопла чи столу, силу обдуву, рівень друкуючої голови відносно столу, швидкість, подачу, поставити на паузу, відмінити друк, тобто всі функції, які доступні через енкодер стануть доступні у програмному застосунку, це вже дає раніше недоступні можливості контролю за роботою апарату.

Камери для міні-ПК бувають різні, від стандартних до більш якісних з більшим числом мегапікселів, або широкоформатні та з нічним баченням.

Ось деякі з них:

- Камера для Raspberry Pi: Оригінальна камера розроблена Raspberry Pi Foundation. Вона має роздільну здатність 5 Мп і може записувати відео з роздільною здатністю 1080p/30fps, 720p/60fps і 640x480p/90fps. Кут огляду становить 67 градусів.
- Інфрачервона камера для нічної зйомки Raspberry Pi NoIR: Це та сама камера, що й оригінальна камера, але без ІЧ-фільтра. Вона може використовуватися для зйомки в умовах слабкого освітлення.
- Raspberry Pi Camera Board v2 - 8 Мегапікселів: Це покращена версія оригінальної камери, яка має роздільну здатність 8 Мп і може записувати відео з роздільною здатністю 1080p/30fps, 720p/60fps та 640x480p/90fps. Кут огляду становить 79 градусів.

- Ширококутна камера Raspberry Pi: Це камера, яка має кут огляду 160 градусів і може використовуватися для зйомки в широкому діапазоні.
- Камера Arducam: Це камера, яка має роздільну здатність 5 Мп і може записувати відео з роздільною здатністю 1080p/30fps. Вона також має кут огляду 160 градусів.
- Камера Waveshare: Це камера, яка має роздільну здатність 5 Мп і може записувати відео з роздільною здатністю 1080p/30fps. Вона також має кут огляду 160 градусів.
- Камера Adafruit: Це камера, яка має роздільну здатність 5 Мп і може записувати відео з роздільною здатністю 1080p/30fps. Вона також має кут огляду 160 градусів.

Всі вони легко підключаються через власний чи окремий шлейф, мають кріплення, а також підключаються програмно в самому міні-ПК, якщо не передбачено, наявні програмні рішення які додадуть можливість підключення.

Raspberry Pi працює на операційній системі Raspbian нині “Raspaberry Pi OS”, яка спеціально розроблена для цього пристрою. Вона включає безліч корисних інструментів і програм, які можуть бути використані для управління 3D-принтером та інших задач.

Raspaberry Pi OS – це безкоштовна операційна система, яка базується на дистрибутиві Debian Linux і оптимізована для апаратного забезпечення Raspberry Pi.

Raspbian була спочатку розроблена у 2012 році як неофіційний порт “Debian wheezy armhf” для Raspberry Pi. Вона була створена для використання на Raspberry Pi і має безліч корисних інструментів та програм, які можуть бути використані для керування 3D-принтером та іншими завданнями.

Raspbian поставляється з більш ніж 35 000 пакетами, попередньо скомпільованим програмним забезпеченням, запакованим у зручний формат для легкої установки на Raspberry Pi. Raspbian також має власне середовище робочого столу PIXEL, яке базується на LXDE і має унікальну тему.

Raspbian була перейменована в Raspberry Pi OS в 2020. Вона є офіційною операційною системою для Raspberry Pi та підтримується Raspberry Pi Foundation.

Raspberry Pi OS встановлюється на всі Raspberry Pi.

Після встановлення операційної системи, можна керувати Raspberry Pi за допомогою клавіатури, миші та монітора, підключених до нього. Також можна керувати Raspberry Pi віддалено через мережу.

Для цього потрібно налаштувати мережеве підключення на Raspberry Pi та підключитися до нього з іншого комп'ютера.

Перший варіант підключення – примітивний.

Для ознайомлення і знання основ можна використовувати SSH-клієнт, щоб підключитися до Raspberry Pi за протоколом SSH і керувати ним через командний рядок.

SSH або Secure Shell (що в перекладі означає «безпечна оболонка») — це мережевий протокол, який використовується для підключення до віддалених комп'ютерів та керування ними за допомогою технології тунелювання. Якщо у вас, наприклад, є сервер в “Timeweb” під керуванням Linux, то ви, напевно, підключаєтеся до нього через OpenSSH (серверна реалізація Secure Shell з відкритим вихідним кодом). Тобто спочатку вводите типову команду в `ssh root@192.168.60.55` і потім виконуєте команди, пов'язані безпосередньо з ОС. Подібні можливості дають технології Telnet та rlogin, але вони не особливо прижилися.

Ключова перевага SSH, порівняно з конкурентами, полягає у підвищеній безпеці. Цей протокол шифрує команди, що передаються, і захищає з'єднання між адміністратором і сервером від третіх осіб.

SSH-клієнт - це програмне забезпечення, яке дозволяє вам підключатися до віддаленого сервера за допомогою протоколу SSH. Це дозволяє виконувати команди на віддаленому сервері, керувати файлами та виконувати інші дії.

Для Raspberry Pi можна використовувати різні SSH-клієнти, такі як Bitvise SSH Client1 або PuTTY2. Обидва ці клієнти є безкоштовними та мають багато корисних функцій.[A]

SSH-клієнт для Windows

Почнемо з найпопулярнішої платформи. Незважаючи на відсутність вбудованих інструментів та загальну неадаптованість під розробку та роботу з серверами, для Windows створили щонайменше десяток функціональних та швидких SSH-клієнтів.

PuTTY

Найвідоміший SSH-клієнт для Windows. Мабуть, єдине, що на слуху у всіх вебмайстрів. PuTTY відрізняється від конкурентів логічним інтерфейсом разом із багатим арсеналом можливостей, включаючи налаштування проксі-серверів та збереження параметрів підключення.

PuTTY поширюється безкоштовно та має відкритий код. При цьому є одним з небагатьох SSH-клієнтів, які досі активно розвиваються і отримують нові версії. Утиліта підтримує протоколи SCP, SSH, rlogin та Telnet, а також працює з усіма методами шифрування даних.

Оригінальна програма доступна лише для Windows, але є порти від спільноти під інші платформи.

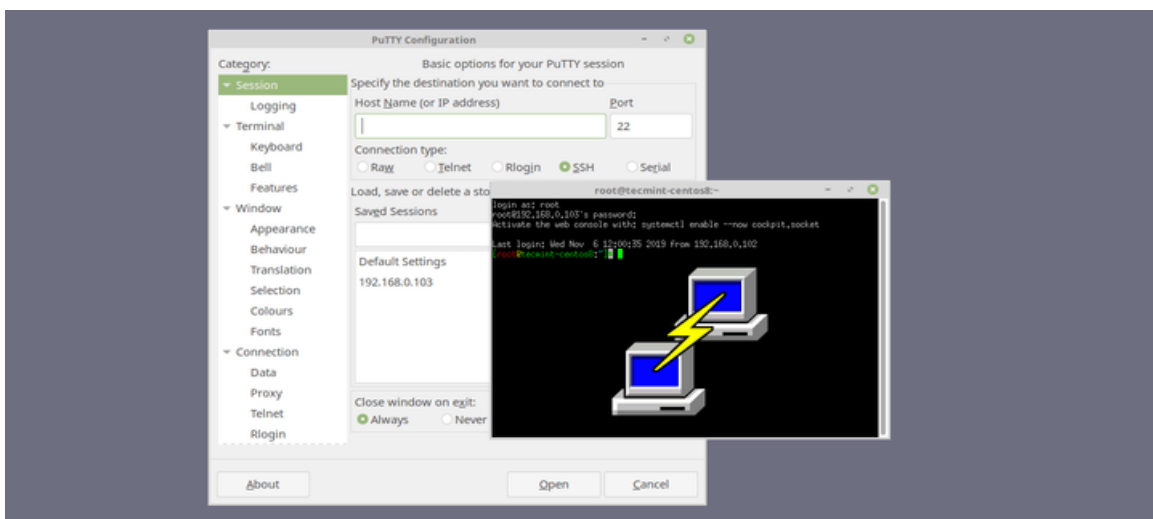


Рис. 1.2. SSH-клієнт для Windows

MobaXterm

Багатофункціональний SSH-клієнт, що полюбився користувачам за високу швидкість роботи, комфортний інтерфейс та купу додаткових функцій, які відсутні у конкурентів. У ньому є браузер файлів, вбудований XServer для управління графічним інтерфейсом на віддаленому комп'ютері, маса плагінів, що розширюють можливості клієнта, та портативна версія, що працює без установки.

Проект умовно-безкоштовний, тому більшість функцій недоступна до оплати. Якщо не купувати платну версію, то функціональність MobaXterm мало чим відрізнятиметься від такої в PuTTY. За професійну версію доведеться віддати 69 доларів.

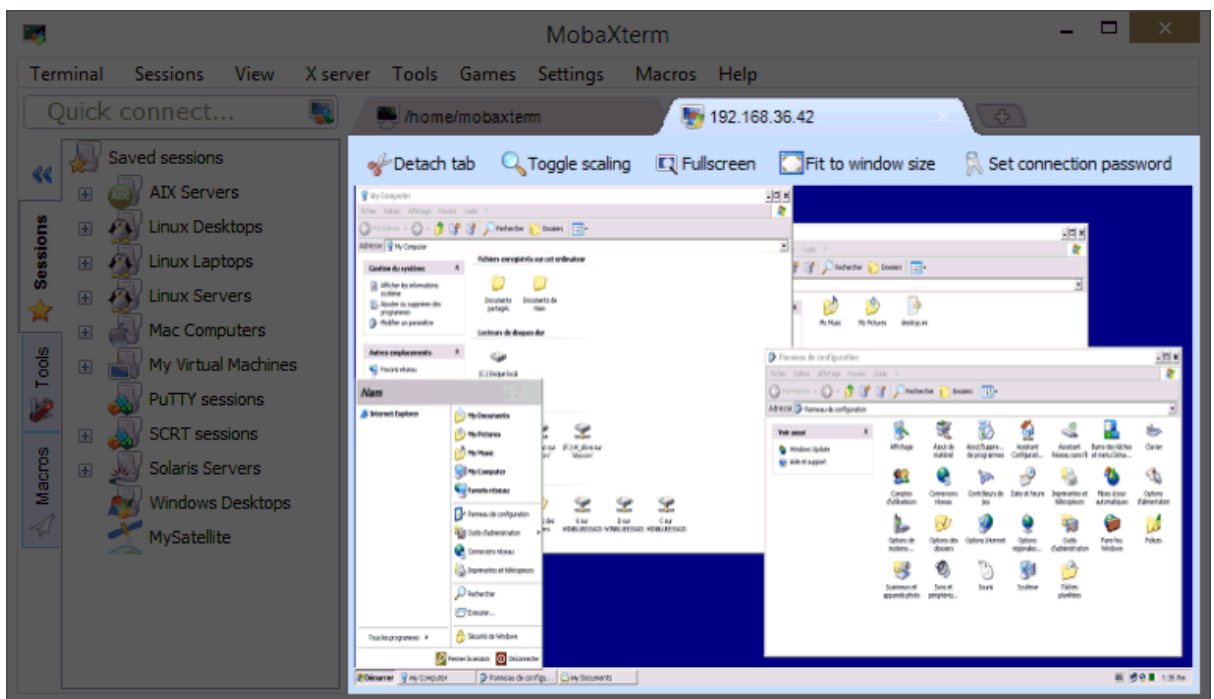


Рис. 1.3. Багатофункціональний SSH-клієнт

Слід зазначити, що більшість користувачів поступить з платною версією точно так само як і з платною Windows, знехтувавши ліцензією.

Другий спосіб. Віддалене керування через веб-інтерфейс.

Другий більш корисний і зручний варіант, використовувати спеціальні програми або операційну систему яка здатна керуватись через інтернет.

Щоб керувати Raspberry Pi через веб-інтерфейс, слід встановити програмне забезпечення, таке як VNC Server чи Apache Guacamole. Це дозволить керувати Raspberry Pi через веб-браузер на іншому комп'ютері.

VNC Server - це програмне забезпечення, яке дозволяє дистанційно керувати пристроями з будь-якого місця. Можливо завантажити VNC Server безкоштовно з RealVNC і отримати повний досвід VNC® Connect з VNC Viewer, безпекою, надійністю та більшими функціями. Інша альтернатива - це UltraVNC, який також є безкоштовним і відкритим джерелом.

VNC Server може керувати Raspberry Pi 3. Щоб дистанційно керувати Raspberry Pi 3, потрібно налаштувати веб-інтерфейс.

Є кілька варіантів веб-інтерфейсів для керування Raspberry Pi 3 B.

Один з них - це AranaCorp, який дозволяє створювати власний веб-інтерфейс для керування Raspberry Pi.

Інший варіант - це TechRepublic, який надає детальну інструкцію щодо налаштування Raspberry Pi 3 Model B. Ця інструкція включає встановлення операційної системи на microSD-карту та налаштування Raspberry Pi B+ за допомогою комп'ютера з Windows.

Apache Guacamole - працює без необхідності встановлення, його суть в тому, що це веб-сервіс, який дає доступ до робочого столу Raspberry OS.

Ще один спосіб - створити власний веб-інтерфейс за допомогою Python та Flask. Flask - це мікрофреймворк Python, який дозволяє легко створювати веб-додатки. Можливо використовувати Flask для створення власного веб-інтерфейсу, який дозволить вам керувати Raspberry Pi через веб-браузер.[\[A\]](#)

Після детального вивчення було виявлено, що дані способи є напів мірою, потребують доопрацювання під задачі автоматизації та не підходять для звичайного користувача, також це не рентабельно.

Був знайдений кращий варіант, повноцінне рішення, спеціально налаштована операційна система для ARM комп'ютерів, з упором під роботу з 3D принтером, але для цього спочатку встановлюється ОС Raspberry Pi OS, а вже потім серверне програмне забезпечення, або одразу готовий варіант операційної системи.

OctoPrint - це безкоштовний веб-додаток з відкритим кодом, який дозволяє керувати та моніторити 3D-принтер та друковані завдання з браузера. Дає можливість отримати доступ до відеопотоку веб-камери, візуалізатора GCODE, рівняння ліжка, оновлень прошивки, плагінів та іншого.[A]

OctoPrint дозволяє керувати та моніторити кожний аспект 3D-принтера та друкованих завдань прямо з браузера. Можливість контролювати температуру нагрівальних елементів та ліжка, переміщувати голову друку по всіх осях, заправляти філамент, витягувати філамент або визначати свої власні користувацькі елементи керування. Запускати, зупиняти або просто призупиняти поточне друковане завдання, контроль процесу у будь-який момент.

Також якщо буде відсутнє інтернет з'єднання, принтер просто продовжує працювати за програмою.

OctoPrint сумісний з більшістю доступних споживчих 3D-принтерів. Його потужна система плагінів дозволяє розширювати його функціональність різними способами. Наприклад, можливо візуалізувати рівень ліжка за допомогою Bed Level Visualizer, створювати захоплюючі таймлапси, які здається, що просто виростають з ліжка друку за допомогою Octolapse, повністю контролювати тематику інтерфейсу користувача OctoPrint за допомогою Themeify, легко оновлювати прошивку принтера прямо з OctoPrint за допомогою Firmware Updater, додавати підтримку для конкретних принтерів, таких як старі принтери Flashforge або старі Makerbots або поточні машини Flashforge, Dremel або PowerSpec.

Офіційний репозиторій плагінів OctoPrint інтегрований прямо в OctoPrint, а встановлення плагіна зводиться лише до короткої інструкції. І початок розробки плагінів дуже простий.

OctoPrint є безкоштовним програмним забезпеченням з відкритим кодом, що випущено під ліцензією “GNU Affero General Public License (AGPL)”. Весь його вихідний код доступний у його репозиторії GitHub, і кожен може модифікувати його на свій розсуд.

OctoPrint може керувати 3D-принтером через Raspberry Pi 3.

Щоб використовувати OctoPrint з Raspberry Pi, одразу слід буде встановити OctoPi на Raspberry Pi.

OctoPrint можливо не встановлювати на Raspberry Pi, а одразу встановити OctoPi, який є образом для скачування на SD-карту яка вставляється у Raspberry Pi, таким чином Raspberry Pi Os вже містить OctoPrint та все необхідне для його запуску.[\[A\]](#)

OctoPrint та його можливості:

MJPEG-Streamer для прямого перегляду друку та створення таймлапсів, сумісних з різними USB-веб-камерами та камерою для Raspberry Pi.

Рекомендоване апаратне забезпечення: Raspberry Pi 3B, 3B+, 4B та новіші моделі.

Була знайдена складність використання у працюючому принтері під керівництвом OctoPi, під час трансляції та роботи штучного інтелекту одночасно, вони здатні перевантажити систему та створювати артефакти і дефекти друку.

Головною причиною є принцип роботи принтеру через OctoPi.

Після створення файлу для принтеру формату GCode, він не завантажується у внутрішню пам'ять принтеру, вона відсутня.

Замість цього він передає команди з файлу до процесів принтеру, постійно, частинами. В цьому випадку неполадки у роботі міні-ПК здатні перервати процес друку, або створити дефекти, якщо команди не будуть відправлятися завчасно то принтер почне робити паузи через відсутність команд, і створювати шароподібні напливи або видати критичну помилку, в такому випадку потрібен дійсно потужний Raspberry, що призводить до збільшення бюджету.

Очікуються артефакти друку та тривалий час завантаження з іншими параметрами, особливо під час додавання веб-камери або встановлення сторонніх плагінів. Звісно також, що налаштування без рекомендованого обладнання офіційно не підтримуються. Raspberry Pi Zero та Zero W явно не рекомендовані взагалі, оскільки спостерігаються серйозні проблеми з продуктивністю, спричинені інтерфейсом WiFi, коли використовується пропускна здатність (наприклад, веб-камера веде трансляцію), що негативно впливає на якість друку.

В даному випадку є можливість не навантажувати Raspberry відправкою команд з файлу, а залишити це на внутрішній процесор самого принтеру.

Перший спосіб, завантажити файл на Micro-SD карту, з якого принтер буде зчитувати команди друку.

Постає проблема відсутності автоматизації в даній частині, OctoPi не має доступу до даних на карті пам'яті, тому він не здатен записати на неї файл, це доведеться робити вручну.

Є технологічний вихід з ситуації: купити SD карту з модулем Wi-Fi.

На жаль, відсутня Micro-SD карта з модулем бездротової передачі, а лише великоформатна SD та перехідник з модулем, в самому принтері наразі встановлений порт для Micro-SD тому доведеться купити перехідник з SD на Micro-SD, що вирішить проблему, але створить деталь яка виділяється.

Після цього підключивши карту пам'яті до Wi-Fi вона стане доступною до відправки файлів по мережі.

Наступна проблема у неможливості підключення OctoPi до Wi-Fi карти пам'яті через відсутність функціоналу, упущення зі сторони розробників ПО.

Користувачі вже створили способи вирішення проблем.

"OctoPrint-WifiSdSupport" – додає підтримку Toshiba FlashAir Wi-Fi SD-карт. Цей плагін автоматично копіює кожен файл, завантажений до OctoPrint, на SD-карту через Wi-Fi.

Таким чином купивши карту пам'яті від фірми Тошіба, і встановивши плагін, з'явиться можливість оптимізувати роботу, шляхом розділення обов'язків між процесорами.

Сам OctoPrint встановлюється з офіційного сайту, також треба скачати **Raspberry Pi Imager** і вже в ньому провести налаштування образу, зокрема ввести дані бездротової мережі. Після чого встановлюється образ на міні-ПК і віддалений доступ стане доступний через веб сервіс.

Після підключення міні-ПК та налаштування віддаленого доступу з вмінням модифікувати веб інтерфейс і Raspberry Pi OS, починається етап розширення можливостей системи 3D друку.

Розробка, створення та підключення маніпулятора.

Робот-маніпулятор – механізм рухомий електродвигунами та керований за допомогою спеціальних команд які віддає контроллер, або комп'ютер віддає команди контролеру.

Маніпулятор має всім знайому конструкцію, з основи яка надійно закріплена на столі чи самому принтері, йде направляюча з механізмом на кінці поєднаний з наступним звеном руки, просторові механізми з багатьма ступенями свободи поєднані у рухому структуру, де кожним плечем рухає електродвигун.

Наприклад маніпулятор з 5 направляючих має від 4 моторів у зв'язку з передачею, 5-ий може бути в основі для більшої рухомості, а 6-ий у клешні яка здатна хапати модель знімати її з платформи, і ставити у визначене місце.

Мотор і передача шестернями здатна перетворити велику кількість обертів двигуна на малу кількість з великим крутним моментом, отримавши більшу силу замість швидкості обертання.

Є маленький електродвигун з обертами у 1000 за хвилину і силою у ньютон метрах, яку можна записати як одиницю сили, якщо ми до ротору двигуна додамо планетарну або іншу передачу з співвідношенням 1 до 2, то ми отримуємо 500 обертів на хвилину і силу значення 2 як вихідну потужність з передачі під час роботи двигуна.

Цікавий факт полягає у тому, що двигун під'єднаний до контролеру здатний перетворювати рух в оберненому порядку, тоді електродвигун стає генератором і здатен пошкодити під'єднаний до нього контролер, слід уникати рухів частин маніпулятора вручну під час його роботи.

Необхідність розробляти маніпулятор спочатку – відсутня. Інтернет має велику базу проектів типу "Open Source", основу можна запозичити та вдосконалити, збільшивши кількість осей руху, змінивши пропорції, додати механізм зміни насадки, тощо.

Термін "6DOF" означає "шість ступенів свободи" (Six Degrees of Freedom) і використовується в різних областях, включаючи робототехніку, аерокосмічну

інженерію, і віртуальну реальність. Ця концепція виходить з фізики і описує можливість об'єкта переміщатися в тривимірному просторі.

В робототехніці 6DOF зазвичай означає, що робот або маніпулятор може переміщатися і орієнтуватися у шести незалежних напрямках. Ці шість ступенів свободи включають три лінійні переміщення (вверх/вниз, вліво/вправо, вперед/назад) і три кутові обертання (поворот вліво/вправо, нахил вперед/назад, нахил по боках).

Шість ступенів свободи в робототехніці дозволяють маніпуляторам точно виконувати складні завдання в різноманітних середовищах, імітуючи багато аспектів людської руки та зап'ястка. Це забезпечує високу гнучкість і точність в різних застосуваннях, від виробничих ліній до медичних досліджень.

Перший приклад це **6DOF Robotic Arm**.

Даний апарат підлягає модифікації і доволі дешевий, з легкістю можна додати механізм для зміни насадок і він зможе знімати модель лезом чи хапати її клешнею.



Рис. 1.3. Робот-маніпулятор 6DOF від BillyJo182.

Можна роздрукувати деталі , приєднати сервоприводи та почати з ним працювати. Рука становить приблизно 35 см. Також налаштовано кріплення сервоприводів для повороту та нахилу, щоб виключити потребу в гайках і болтах, лише самонарізні гвинти. Це набагато легше, ніж боротися з крихітним обладнанням для монтажу сервоприводів.

Дешевий варіант, легкий і примітивний, однак повинен справлятися із задачею підбирання невеликої моделі, після модифікації буде здатен змінювати насадку на шпатель для відклеювання моделі від платформи.

Основне керування лягає на **Arduino**.

Робот-маніпулятор на базі Arduino працює за допомогою комбінації апаратного та програмного забезпечення.

Можна замінити на мікроконтролер для Raspaberry.

Arduino плата – це мікроконтролер, який служить умовно мозком робота, скоріше командним пунктом який приймає команди від іншого пристрою. Arduino може бути різних моделей, наприклад, Arduino Uno, Mega, Nano або інші.

Сервоприводи на основі крокових безколекторних двигунів або степери(BLDC), використовуються для точного керування рухами маніпулятора, але повільного і дорожчого ніж на іншому типі двигунів. Такі мотори масивні, дорогі, та потребують “драйвер”, плата або частина плати яка має чотири виходи, і повністю контролює кількість обертів двигуна. Використовуються для роботи кінематики самого принтеру, не слід використовувати у роботі-маніпуляторі.

Сервоприводи на основі колекторних двигунів, мають просте живлення від прямого струму, і керуються за допомогою розрахунку необхідного часу на рух, на відміну від крокових, можуть працювати від двох проводів. Дешеві, та зручні, саме їх використовують у більшості випадків.

Сервоприводи можуть точно контролювати положення, швидкість та кут повороту окремих частин маніпулятора, завдяки пошаговій роботі драйверу для кожного мотору по черзі або всі одночасно, комп'ютер може визначити кількість скачків чи час навантаження для кожного двигуна.

Використовуючи мову програмування Arduino (похідну від C++), створюється програма, яка контролює поведінку маніпулятора. Програма визначає, як і коли рухати сервоприводами, враховуючи конкретні задачі маніпулятора.

```

1 #include <Servo.h>
2
3 // Створення об'єктів сервопривода
4 Servo servo1;
5 Servo servo2;
6 Servo servo3;
7
8 void setup() {
9     // Приєднання сервоприводів до відповідних пінів
10    servo1.attach(9);
11    servo2.attach(10);
12    servo3.attach(11);
13
14    // Ініціалізація сервоприводів у вихідне положення
15    servo1.write(90);
16    servo2.write(90);
17    servo3.write(90);
18 }
19
20 void loop() {
21     // Тут можуть бути алгоритми для читання сенсорів, прийняття рішень
22     // і відправлення команд сервоприводам для виконання рухів.
23
24     // Наприклад, просте переміщення сервопривода від 0 до 180 градусів:
25     for (int pos = 0; pos <= 180; pos += 1) {
26         servo1.write(pos);
27         delay(15); // Чекаємо 15 мс для плавного руху
28     }
29
30     for (int pos = 180; pos >= 0; pos -= 1) {
31         servo1.write(pos);
32         delay(15); // Чекаємо 15 мс для плавного руху
33     }
34
35     // Додати логіку для інших сервоприводів і завдань маніпулятора
36 }

```

Рис. 1.4. Програма контролю поведінки маніпулятора

Підключення на налаштування сервоприводів

Робот-маніпулятор потребує джерела живлення, яке забезпечує електроенергією як саму плату Arduino, так і сервоприводи. В такому випадку джерело живлення повинно мати потужність не меншу ніж загальна потужність сервоприводів. Після аналізу стало ясно, що мотори навіть просто підключені споживають струм у простої, 180 мА при 6 В. Під час роботи цей струм зростає у рази по мірі збільшення навантаження, струм може зрости до 2 чи 3 ампер, в такому випадку мотор може споживати велику кількість струму, яка буде в районі 10 чи навіть 15 ватт. Мотор нагрівається під час роботи, під навантаженням можливий перегрів і вихід з ладу, краще замінити на більш продвинуті чи модифікувати корпус отворами при можливості для кращого охолодження.

Якщо рахувати максимальну потужність двигунів як 7 по 10 ватт, виходить Блок живлення повинен мати вихід у 5 вольт та 14 ампер або загалом 70 ватт, так як більший вольтаж можливий.

Є можливість використовувати потужний блок живлення для телефону, комп'ютерний блок живлення чи куплений спеціально.

Для складніших задач можуть бути використані датчики, які збирають інформацію про оточення або стан маніпулятора. Це можуть бути датчики тиску, ультразвукові датчики відстані, гіроскопи, сенсори тощо.

Під час роботи програма на Arduino відправляє сигнали до контролеру, керуючим сигналом для подачі струму на мотор не пропускаючи потужність блоку живлення через себе до сервоприводів, керуючи їхньою позицією та рухом. Це може бути як простий рух (наприклад, поворот на певний кут), так і більш складні сценарії, що включають послідовність рухів для виконання конкретних задач.

Компоненти Arduino:

Мікроконтролер: Ядро Arduino - це мікроконтролер (зазвичай від Atmel). Наприклад, Arduino Uno використовує ATmega328.

Вхідно-вихідні (I/O) порти: Плати Arduino мають набір аналогових та цифрових I/O пінів, які можна використовувати для підключення різних зовнішніх пристроїв, від датчиків до моторів.

Живлення: Arduino можна живити через USB-порт від комп'ютера або через зовнішнє джерело живлення.

Інтерфейси з'єднання: Arduino має різні порти для з'єднання з комп'ютером (зазвичай через USB) для програмування, а також для з'єднання з іншими пристроями та модулями.

Інтегроване середовище розробки (IDE): Arduino програмується за допомогою спеціального IDE, яке можна завантажити безкоштовно. Це середовище підтримує мову програмування, що базується на C/C++.

Написання скетчів: Програми для Arduino називаються "скетчами". Вони складаються з двох основних функцій: `setup()` (виконується один раз при старті програми) та `loop()` (циклічно повторюється). Завантаження програми: Після написання скетча його можна завантажити на плату через USB-підключення.

Фізичне підключення: Arduino можна підключити до Raspberry Pi через USB-порт. Це забезпечить як електропостачання, так і можливість передачі даних.

Комунікація між пристроями: Можна використовувати різні протоколи для комунікації між Raspberry Pi та Arduino, наприклад, через Serial порт.

Програмування Raspberry Pi: Для взаємодії з Arduino з Raspberry Pi, можна використовувати Python або інші мови програмування, які підтримують серійне з'єднання.

Інтеграція систем: Зазвичай Raspberry Pi використовується для обробки більш складних задач, як от мережеві комунікації, обробка даних або веб-сервер, тоді як Arduino керує більш простими апаратними взаємодіями.

Ця конфігурація дозволяє використовувати переваги обох платформ: гнучкість та простоту Arduino для апаратного управління та більшу обчислювальну потужність Raspberry Pi для більш складних операцій.

Serial порт, також відомий як послідовний порт, - це інтерфейс для обміну даними між пристроями.

У контексті Arduino та Raspberry Pi, послідовний порт часто використовується для комунікації між цими двома пристроями.

Передача даних: Serial порт дозволяє передавати дані один за одним (послідовно), а не паралельно. Це означає, що дані передаються по одному біту за раз через один або два проводи.

TX та RX: У простих послідовних з'єднаннях є два основні контакти: TX (transmit, передача) та RX (receive, прийом). TX одного пристрою підключається до RX іншого, і навпаки.

У Arduino, послідовний порт використовується для завантаження скетчів з комп'ютера, а також для комунікації з іншими пристроями, наприклад з Raspberry Pi.

Arduino IDE має вбудований серійний монітор, який дозволяє відправляти та отримувати текстові дані через Serial порт.

З'єднання Arduino з Raspberry Pi через Serial порт:

Підключення: Для з'єднання Raspberry Pi з Arduino через Serial порт, ви підключаєте TX пін Arduino до RX піна на Raspberry Pi, і RX пін Arduino до TX піна на Raspberry Pi.

Програмування комунікації: На Raspberry Pi можна використовувати мову програмування - Python, з бібліотеками, які підтримують серійну комунікацію (pySerial). На Arduino слід використовувати вбудовані функції Serial для обміну даними.

Обмін даними: Після встановлення з'єднання, Arduino та Raspberry Pi можуть обмінюватися даними через Serial порт, використовуючи прості команди для відправлення та отримання даних. Цей метод забезпечує надійний спосіб комунікації між Arduino та Raspberry Pi для різних проектів, від простого обміну даними до складних інтегрованих систем.

Цікавий сам механізм роботи Arduino. Arduino управляє роботом-маніпулятором шляхом відправлення сигналів керування до сервоприводів або степперних двигунів, які забезпечують рух механічних частин маніпулятора.

Сервоприводи - це тип двигунів, які можуть точно керувати положенням. Кожен сервопривод має вхід для керуючих сигналів, який підключається до одного з цифрових виходів Arduino.

Arduino генерує ШІМ-сигнали (широтно-імпульсну модуляцію) для керування кутом повороту сервопривода. Довжина імпульсу сигналу визначає, на який кут повернеться вал сервопривода.

Крокові двигуни рухаються ривками, що дозволяє точно контролювати їхнє положення.

Використовується серія електричних імпульсів для визначення кількості та напрямку кроків, що робить кроковий двигун.

Саме програмування полягає у тому, що ви пишете скетч на Arduino, який визначає логіку керування маніпулятором. Це може включати читання даних з датчиків, розрахунок траєкторій руху, та генерацію відповідних сигналів для двигунів.

Наприклад, скетч може включати код для прийняття команд з зовнішнього джерела (наприклад, з комп'ютера або іншого мікроконтролера) та відправлення відповідних сигналів до сервоприводів або степперних двигунів для виконання цих команд.

Також живлення має умовні труднощі.

Сервоприводи та крокові двигуни часто вимагають більше енергії, ніж може надати Arduino, тому вони підключаються до зовнішнього джерела живлення. Але сигнали керування все одно надходять від Arduino.

Сам маніпулятор складається з різних механічних частин, які рухаються за допомогою сервоприводів або степперних двигунів.

Використовуючи цей принцип, Arduino може управляти складними рухами робота-маніпулятора, такими як вибір і переміщення предметів, точне позиціонування частин маніпулятора, та навіть виконання заданих сценаріїв взаємодії з об'єктами або середовищем.

Може скластися думка, що Raspberry просто дає умовну команду, наприклад, повернути сервопривід на 10 градусів, а Arduino просто включає живлення двигуна на певний момент, але це не так.

Raspberry Pi відправляє специфічну команду до Arduino, наприклад, через Serial порт. Ця команда може включати інформацію про те, на який кут необхідно повернути сервопривід.

Arduino отримує цю команду та обробляє її у своєму скетчі (програмі). Програма визначає, як потрібно відреагувати на команду - у цьому випадку, як повернути сервопривід на вказаний кут.

Для керування сервоприводом, Arduino генерує ШІМ-сигнал (широтно-імпульсну модуляцію). Цей сигнал контролює кут повороту сервопривода. Довжина імпульсу ШІМ визначає, на який кут повернеться вал сервопривода. Керування сервоприводом: Сервопривід отримує ШІМ-сигнал від Arduino та повертає свій вал на відповідний кут. Це не просто включення або виключення живлення, а точне керування положенням вала двигуна.

Таким чином, Arduino діє як посередник, який отримує високорівневі команди від Raspberry Pi, перетворює їх у низькорівневі сигнали для сервоприводів та забезпечує точне керування механічними частинами маніпулятора. Це дозволяє створювати складні та точні системи управління для різноманітних застосувань.

Робот маніпулятор – **6DOF Robotic Arm** готовий проект до якого слід підключити плату керування.

За необхідності можна втрутитись у саму модель перед друком, змінивши розмір, форму і оформлення деталей корпусу, змінивши зовнішній вигляд і призначення, переробивши конструкцію повністю під свої побажання.

Однак, після вивчення варіантів маніпуляторів для домашнього використання, я знайшов інший подібний проект, який на відміну від попереднього новий та розвинутий.

Даний робот розповсюджується у форматі широкого доступу, тому кожен може скачати 3д модель роботу. На жаль, відсутня інструкція по підключенню робота до Arduino Mega, яка зазначена в описі, та відсутня клешня, вона інтегрується окремо.



Рис. 1.5. 6DOF ROBOT ARM від dannyvandenheuvel.[5]

Присутній сам корпус робота та корпус для апаратного забезпечення, що достатньо з механічної точки зору, так як вже передбачені всі кріплення та пази сервоприводів.

Може бути використано як основу для власного маніпулятора, після його друку та налаштування інтерфейсу керування.

Після нової необхідності у підключенні майбутнього робота до апаратного забезпечення, такого як Arduino, продовжені дослідження по темі привели до корисного аналогу, який вирішив деякі проблеми.

Проект на **GitHub** називається "6 DOF robot arm controlled through a Raspberry Pi" і включає в себе підключення сервоприводів до Raspberry Pi за допомогою драйвера PWM PCA9685.[6]

Таким чином був знайдений варіант, який складніший з точки зору програмування, але дешевший і більш потужний, так як буде більше можливостей для програмування і менша затримка, завдяки чому отримані команди будуть йти від самого міні-комп'ютера, але збільшиться навантаження на процесор, тому можливо доведеться використовувати новіший Raspberry Pi 4.

Драйвер PWM PCA9685 - це 16-канальний 12-бітний драйвер для управління сервоприводами, вироблений компанією Adafruit[7]. Цей драйвер використовується для управління багатьма сервоприводами одночасно, особливо коли використовується мікроконтролер з обмеженою кількістю виводів PWM, такий як Raspberry Pi.

Програмований через програмне забезпечення, PCA9685 дозволяє всім або певним групам пристроїв PCA9685 реагувати на загальну адресу шини I2C, що мінімізує команди I2C-шини. Має 6 апаратних адресних контактів, що дозволяє підключати до 62 пристроїв до однієї шини I2C.[8]

Особливості PCA9685 включають:

- Управління до 16 сервоприводів або світлодіодів.
- Інтерфейс I2C для зв'язку з мікроконтролером.

- Відокремлене живлення для сервоприводів, що дозволяє використовувати вищі напруги та струми, ніж може надавати мікроконтролер.

- Вбудована підтримка модуляції ширини імпульсу (PWM), яка необхідна для управління сервоприводами.

- Використання цього драйвера з Raspberry Pi дозволяє створити більш складні і потужні системи управління роботизованими механізмами, такими як 6-осьові роботизовані маніпулятори.

Для програмування контролера PCA9685, під час використання Raspberry Pi з веб-інтерфейсом на ОС OctoPi.

Основні кроки включають:

1. Підключення маніпулятора до Raspberry Pi:

Підключити драйвер PCA9685 до Raspberry Pi через I2C-шину.

Для цього з'єднати SDA (дані) і SCL (годинник) на PCA9685 з відповідними пінами на Raspberry Pi.

Також з'єднати землю (GND) і джерело живлення (VCC) з Raspberry Pi.

Підключити сервоприводи маніпулятора до каналів PCA9685.

2. Встановлення та налаштування OctoPi:

Встановити OctoPi на Raspberry Pi через карту пам'яті.

Увімкнути і налаштувати I2C на Raspberry Pi через "raspi-config".

3. Розробка плагіну для OctoPrint:

Розробити плагін для OctoPrint, який дозволить створити інтерфейс для керування сервоприводами маніпулятора.

В плагіні додати функціонал для створення кнопок в інтерфейсі OctoPrint, які будуть відповідати за рух кожного мотора маніпулятора.

4. Програмування логіки управління:

Написати скрипт Python, який буде приймати команди від плагіна OctoPrint і передавати сигнали на PCA9685 для управління сервоприводами.

У скрипті визначити функції для руху кожного сервоприводу відповідно до команд від інтерфейсу.

5. Тестування та калібрування:

Протестувати кожен сервопривід, використовуючи плагін і скрипт, щоб переконатися, що всі мотори працюють коректно.

Провести калібрування, якщо необхідно, для точного керування маніпулятором.

6. Інтеграція з OctoPrint:

Встановити та активувати плагін в OctoPrint.

Використати інтерфейс OctoPrint для керування моторами маніпулятора через веб-інтерфейс.

1.9. Система безперебійного живлення

Ми існуємо в реаліях, які привертають увагу до можливих проблем з електропостачанням у скрутні часи.

Сьогодні показало, що електроживлення будинку може бути перервано в будь-який момент. В такому випадку навіть мигаюче світло здатне зупинити процес друку і зіпсувати роботу, інертність і внутрішній запас енергії у конденсаторах та інших елементах блоку живлення не достатній, щоб на секунду підтримувати віддачу постійного струму, і комп'ютер принтеру просто перестає друкувати, відновлення друку може не спрацювати.

Тому можливо запобігти проблемам під час збоїв живлення таких як:

- відключення міні-пк і втрата прогресу,
- помилка відновлення роботи,
- відлипання моделі через відсутність обігріву стола,
- можливі поломки.

Аби запобігти цьому слід впровадити систему безперебійного живлення.[8]

Система має 4 основні компоненти:

- Акумуляторний блок.
- Зарядний пристрій чи джерело зеленої енергетики.
- Автомобільний інвертор 12/220.

- Реле.

Акумуляторний блок

Слід розглядати 2 можливі типи

Перший тип – свинцевий акумулятор

- Дешевий.
- Громіздкий.
- Має низьку пожежну небезпечність.
- Низька щільність зберігання енергії
- Не тривале зберігання енергії.

Звичайний автомобільний акумулятор 12V на 60 ампер-годин, здатен зберігати теоретично – 720Wh.

12 вольт множать на ампер-години і отримують приблизну ємність у ватах, таким чином звичайний акумулятор здатен зберігати до 700 ват, але це максимальне можливе значення і може різко зменшитися.

Враховуючи енергоспоживання принтера на годину, це приблизно 300 ват, виходить що ємності одного акумулятора вистачить лише на годину чи півтори, так як присутні енерговтрати і поріг розряду, який не дасть перетворювачу працювати до кінця ємності.

Відсутність енергій може тривати годинами, слід паралельно під'єднувати акумулятори для збільшення ємності акумуляторного блоку не збільшуючи вольтаж.

Другий тип – літійові акумулятори:

- велика ємність на вагу,
- швидка зарядка,
- довгий заряд,
- висока ціна,
- пожежонебезпека.

В такому випадку купують акумуляторні комірки, спеціальні чи від електромобіля, роблять все теж саме, але потрібне спеціальне обладнання для

зарядки з контролем заряду, однак це правило стосується всіх акумуляторних елементів.

Дорогий варіант здатний зменшити габарити АКБ у 4 рази.

Автомобільний інвертор.

Автомобільний інвертор чи АС/DC перетворювач здатен приймати 12 вольт 24 чи більше і перетворювати прямий ток АКБ у змінний струм 220 вольт модифікованої хвилі.

Слід звернути увагу на потужність такого апарату, часто на них зазначено, наприклад, 2000 ват потужності, насправді це пускова потужність на частину секунди для підключення моторів напругу.

Справжня потужність зазначена у характеристиках і може бути приблизно 500 ват, мінімум для справної роботи.

Реле.

Останньою частиною є реле. Воно здатне переключитись з ліній електропередач будинку на перетворювач менше ніж за секунду.

Таким чином створюється власна система безперебійного живлення.

Огляд аналогічних систем автоматизації.

По тематиці наукової роботи вже наявні деякі роботи, технологічні рішення і навіть продукти.

Розробки з автоматизації процесу друку ведуться різними фірмами, серед них можна виділити один з непоганих прикладів від фірми Formlabs.

Екосистема Автоматизованого фотополімерного 3D друку Formlabs.

Це комплексне рішення і доволі просте, воно дозволяє перетворити 3D принтер типу LFS в автоматизовану систему, яка сама знімає надруковану деталь у спеціальний контейнер, і підтримує запас матеріалу, відслідковуючий ємність баку.

LFS 3D принтер - це скорочення від "Large Format Stereolithography 3D Printer", що означає 3D принтер великого формату на основі стереолітографії.

Стереолітографія - це технологія 3D друку, яка використовує фотополімерну смолу і ультрафіолетове випромінювання для створення твердих об'єктів за цифровою 3D моделлю.

LFS 3D принтери здатні друкувати невеликі об'єкти з високою точністю і якістю поверхні. Деякі приклади LFS 3D принтерів - це Form 3L від Formlabs і ProX 800 від 3D Systems.

Насправді принтери, які використовують не пластик, а так звану смолу застигаючу від ультрафіолетових промінів стають популярними. Все через властивості рідини застигати під дією ультрафіолету, що дає змогу створювати високо деталізовані і точні деталі.

Мінуси:

- більшість смол токсичні;
- дорога сировина;
- менші розміри відносно ціни принтерів;
- довший процес друку.

Такі принтери не витіснять FDM принтера.

4 січня Formlabs представили свою Automation Ecosystem – Екосистему Автоматизації фотополімерного 3D друку. Це комплексне рішення, яке дозволяє перетворити парк LFS 3D принтерів Formlabs Form 3, Form 3+, Form 3B або Form 3B+ на потужну, самостійно працюючу лінію з виробництва друківаних частин з великої бібліотеки матеріалів Formlabs.

Екосистема Автоматизації Formlabs уможливілює цілодобовий 3D друк без участі людини, мінімізує витрати на деталь та економить застосування робочої сили до 80 %.[2]

Фірма створила міні конвеєр, він здатен створювати нові деталі знову і знову, має чергу, має віддалений контроль, звісно, в смоляних принтерах вірогідність дефектів мінімальна.

Насправді зручна система, наявна програма Fleet control:

- Керувати кількома 3D-принтерами через одну централізовану чергу у Dashboard, де можна додавати, дублювати та пріоритезувати завдання на друк у будь-який час.

- Оптимізувати виробництво по всьому парку 3D-принтерів. Коли завантажується завдання на друк Dashboard, Fleet Control автоматично призначає принтер для друку на основі загальних витратних матеріалів.

- Отримувати живі оновлення та сповіщення про виконані друки, принтери та групи принтерів.

- Контролювати права доступу та видимість команди. Ви можете встановити рівні доступу адміністратора та учасника всередині груп принтерів.

- Оновити всі ваші принтери до останньої версії прошивки автоматично або вибрати попередню версію та застосувати її до принтерів зі списку.

- Переглянути ключову статистику про стан вашого парку 3D-принтерів. Ви можете перевіряти статистику використання та обслуговування принтерів.

Fleet Control працює у зв'язці з програмою PreForm, яка служить для підготовки 3D-моделей до друку. Ви можна завантажити PreForm безкоштовно з офіційного сайту Formlabs.

2. ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДО УПРАВЛІННЯ 3D ПРИНТЕРОМ

2.1. Особливості методів машинного навчання для управління 3D принтером

Головною задачею є початок розробки штучного інтелекту з комп'ютерним зором використовуючи інтегрування вільних методів та поліпшення за допомогою машинного навчання .

Необхідно створити нейронну технологію, здатну виконувати конкретні задачі з детекції вад друку для запобігання зайвих витрат, та впровадити систему віддаленого керування принтером, створивши систему з веб інтерфейсом для можливості керування друком.

Додатковою частиною є можливість навчити нейромережу керувати роботом-маніпулятором.

У сучасному світі при його технологічному розвитку доступний перший тип штучного інтелекту, він здатний виконувати лише конкретні задачі яким був навчений. Існують складні та важкі моделі широких можливостей, в даному випадку потрібні лише конкретні задачі пов'язані з аналізом фотографій, тому очікується легка модель.

В форматі дослідження спочатку треба описати загальну суть.

Штучний інтелект (ШІ) – це широке поле комп'ютерних наук, яке імітує людські когнітивні функції, такі як навчання, розуміння, розсудливість, і розв'язування проблем.

ШІ можна класифікувати на кілька типів, в залежності від їх здібностей і функціональності.

Перший тип – налаштований ШІ (Narrow AI).

Налаштований ШІ, також відомий як слабкий ШІ, це тип який спроектований і навчений виконувати конкретну задачу.

Особливості:

Обмежений фокус:

Вони фокусуються на одному конкретному завданні та не мають загального розуміння.

Приклади: Розпізнавання мови, чат-боти, рекомендаційні системи.

Застосування:

Широко використовується в індустрії та повсякденному житті, наприклад, в персональних асистентах типу Siri або Alexa.

Другий тип – загальний штучний інтелект.(General AI)

Загальний ШІ - це концепція машини, яка може розуміти, навчатися і застосовувати свої знання в широкому діапазоні завдань, аналогічно людському інтелекту. Наразі є лише у вигляді концептів та секретних розробок.

Третій тип – сильний ШІ (Strong AI)

Надрозум, який належить до сфери фантастики.

Більша частина нейротехнологій – це прості логічні моделі здатні виконувати лише конкретні передбачені задачі.

Наприклад, ChatGPT, належить до простих, але сильно розвинутих моделей, він не здатний до самонавчання, тому не відноситься до другого рівня ШІ. Складний та перерозвинутий, через що непередбачуваний, але він має багато аналогів, також існує велика кількість доступних простіших ШІ, які підійдуть для обробки зображення та слідування простій передбаченій логіці алгоритму дії у певній ситуації.

Налаштовані штучні інтелекти (ШІ), також відомі як "слабкі ШІ", можна класифікувати за кількома категоріями залежно від їхніх специфічних функцій та призначень.

Ось декілька ключових типів:

1. ШІ для обробки мови:

Застосування: чат-боти, персональні асистенти (наприклад, Siri, Alexa), перекладачі.

Функції: розпізнавання, обробка та генерація мовлення і тексту.

2. ШІ для комп'ютерного зору:

Застосування: розпізнавання обличь, системи безпеки, аналіз медичних зображень.

Функції: інтерпретація та обробка візуальної інформації з зображень та відео.

3. Рекомендаційні системи:

Застосування: електронна комерція, стрімінгові сервіси (наприклад, Netflix, Spotify), персоналізовані новинні агрегатори.

Функції: аналізування попередніх виборів користувачів для рекомендації продуктів або контенту.

4. ШІ для прогнозування та аналізу даних:

Застосування: фінансовий аналіз, прогнозування погоди, аналітика великих даних.

Функції: виявлення закономірностей у великих наборах даних для прогнозування майбутніх трендів або поведінки.

5. Автономні роботи:

Застосування: самокеровані автомобілі, промислові роботи, дрони.

Функції: навігація і взаємодія з фізичним світом, часто у відповідь на датчики та інші входи.

6. ШІ для інтерактивних ігор:

Застосування: відеоігри, тренажери, навчальні симуляції.

Функції: створення реалістичних і адаптивних віртуальних середовищ та персонажів.

Кожен із цих типів інтелектів спеціалізується на певних завданнях і вимагає унікальних алгоритмів та підходів для їхнього вирішення. Налаштовані ШІ відрізняються своєю здатністю виконувати дуже конкретні завдання на високому рівні.

Для автоматизацію процесу друку з використанням машинного навчання слід звернути увагу саме на штучний інтелект для комп'ютерного зору.

7. Комп'ютерний зір.

Комп'ютерний зір як результат машинного навчання відноситься до галузі штучного інтелекту, яка зосереджена на інтерпретації та обробці візуальної інформації зі світу навколо нас. Ця область використовує алгоритми для того, щоб надавати комп'ютерам здатність "бачити" та розуміти вміст зображень та відео.

Основні аспекти ШІ для комп'ютерного зору:

1. Обробка зображень: попередня обробка зображень для покращення якості даних, що включає фільтрацію, корекцію яскравості та контрасту, і виявлення країв.

2. Розпізнавання об'єктів: визначення та класифікація об'єктів у зображеннях або відео. Це може включати розпізнавання обличчя, автомобілів, тварин тощо.

3. Сегментація Зображень: розділення зображення на частини або області для подальшого аналізу, наприклад, виділення об'єктів на передньому плані від фону.

4. Трекінг Об'єктів: слідкування за рухомими об'єктами у відео.

Технології та алгоритми:

1. Згорткові нейронні мережі (CNNs): одна з ключових технологій у комп'ютерному зорі, яка використовується для розпізнавання образів та зображень.

2. Глибоке навчання: автоматизований процес навчання, що включає великі набори даних і нейронні мережі для вдосконалення здатностей системи до обробки зображень.

Застосування:

1. Медична діагностика: аналіз медичних зображень, таких як рентгенівські, МРТ та КТ знімки для допомоги у діагностиці захворювань.

2. Системи безпеки: розпізнавання обличчя та контроль доступу.

3. Автономні транспортні засоби: використання комп'ютерного зору для навігації та виявлення перешкод.

4. Робототехніка: використання візуальної інформації для керування роботами в різних середовищах.

5. Аналіз відео: моніторинг та аналіз відеозаписів для безпеки, маркетингових досліджень або спортивних аналізів.

Виклики та обмеження:

1. Величезні обсяги даних: обробка та аналіз великих наборів даних вимагає значної обчислювальної потужності.

2. Точність: помилки у розпізнаванні та класифікації можуть мати серйозні наслідки, особливо у медичній діагностиці та автономних транспортних засобах.

3. Приватність та етика: збір та обробка візуальних даних викликає питання приватності та етичних міркувань.

Майбутнє комп'ютерного зору:

Розвиток комп'ютерного зору продовжує відкривати нові можливості, від розширеної реальності до розумних міст. Однак, для подальшого прогресу цієї галузі необхідні вдосконалення у сферах алгоритмів, обчислювальної потужності, та вирішення етичних та правових питань.

Згорткові нейронні мережі (ЗНМ), відомі також як конволюційні нейронні мережі, є особливим типом глибоких нейронних мереж, які широко використовуються у задачах комп'ютерного зору.

ЗНМ мають декілька основних типів шарів:

1. Згорткові шари (Convolutional Layers):

Функція: вони використовують фільтри або ядра для сканування вхідного зображення і створення карти активацій, яка виявляє особливості, такі як краї, кути, текстури тощо.

Процес: кожен фільтр згортки перетворює частину зображення (наприклад, 3x3 пікселів) у одне число, що відображає присутність певної особливості в цій області.

2. Пулінгові шари (Pooling Layers):

Функція: вони зменшують розмір карти активацій, зберігаючи лише найбільш важливу інформацію.

Типи: максимальний пулінг (вибір найбільшого значення) і середній пулінг (вирахування середнього значення).

8. Повнозв'язні шари (Fully Connected Layers):

Функція: на завершальному етапі, повнозв'язні шари використовуються для класифікації або іншого виду висновку на основі виявлених особливостей.

Процес роботи:

1. Введення зображення: вхідними даними для ЗНМ є зображення, яке проходить через серію згорткових і пулінгових шарів. Ці шари автоматично виявляють та визначають ключові особливості зображення.

2. Виявлення особливостей: на ранніх шарах мережі виявляються прості особливості, такі як краї та кольори, в той час як глибші шари виявляють більш складні особливості, такі як обличчя або об'єкти.

3. Класифікація: на завершальному етапі повнозв'язні шари використовують виявлені особливості для класифікації зображення в певні категорії.

Особливості:

1. Інваріантність до зсуву: ЗНМ ефективні у виявленні особливостей незалежно від їхнього положення на зображенні.

2. Поділ параметрів: фільтри в згорткових шарах діляться між усіма областями зображення, що зменшує кількість параметрів та обчислювальний об'єм.

Застосування: ЗНМ використовуються у широкому спектрі задач, включаючи розпізнавання обличчя, автономні транспортні засоби, медичну діагностику, системи безпеки та багато іншого. Вони є ключовою технологією у сучасному комп'ютерному зорі, забезпечуючи високу точність та надійність в обробці візуальних даних.

Для використання ШІ з комп'ютерним зором на Raspberry Pi, особливо для виявлення дефектів 3D-друку, потрібно враховувати обмежену обчислювальну потужність цього пристрою. Вибір підходящого ШІ залежить від конкретних вимог до точності, швидкості та типу дефектів, які потрібно виявити. Варіанти та аспекти, які варто врахувати:

2.2. Моделі штучного інтелекту для Raspberry Pi

Детально розглянемо основні моделі машинного навчання, які доречно використати для Raspberry Pi:

1. TensorFlow Lite:

Опис: це легковагова версія TensorFlow, спеціально розроблена для мобільних та вбудованих пристроїв.

Переваги: вона оптимізована для швидкого виконання на пристроях з обмеженими ресурсами, таких як Raspberry Pi.

2. OpenCV:

Опис: бібліотека OpenCV (Open Source Computer Vision Library) широко використовується для задач комп'ютерного зору.

Переваги: підтримує різноманітні алгоритми, включаючи ті, які можуть бути використані для виявлення дефектів.

3. EfficientNetLite:

Опис: легка версія популярної моделі глибокого навчання EfficientNet, оптимізована для використання на пристроях з обмеженими обчислювальними ресурсами, таких як Raspberry Pi.

Переваги: ця модель є ефективною для різних завдань комп'ютерного зору, включаючи класифікацію зображень та виявлення об'єктів, і може бути корисною для виявлення дефектів у 3D-друку.

Важливі аспекти:

1. Обмеження Raspberry Pi: важливо враховувати обчислювальні обмеження Raspberry Pi. Моделі повинні бути достатньо легкими, щоб ефективно працювати на цьому пристрої.

2. Оптимізація моделі: моделі можуть потребувати додаткової оптимізації для підвищення швидкості обробки, включаючи квантизацію та спрощення мережі.

3. Інтеграція з камерою: важливо також оптимізувати процес захоплення та передачі зображень.

Існують кілька проектів, де штучний інтелект використовується для виявлення дефектів у процесі 3D-друку:

1. Printpal: ця компанія з Чикаго запустила програмне забезпечення на основі ШІ для виявлення дефектів у 3D-друку. Програма PrintWatch використовує машинне навчання для реального виявлення дефектів, що з'являються під час 3D-друку, дозволяючи користувачам відстежувати дефекти та приймати проактивні заходи.[1]

2. Ai Build: ця лондонська компанія, яка спеціалізується на ШІ та автоматизованому 3D-друку, впровадила інструмент реального часу для контролю якості. Використовуючи комп'ютерний зір та ШІ, вони автоматично виявляють найпоширеніші проблеми друку, такі як усадка матеріалу та спотворення. 1 2

Метод виявлення дефектів FDM-друку: розроблена методика офлайн-виявлення дефектів FDM-друку, що базується на машинному зорі та згорткових нейронних мережах (CNN). Використовується камера CCD для захоплення поточного шару, а спеціально створена CNN-мережа класифікує дефекти.1

Зокрема, нейронні мережі розроблені для виявлення дефектів 3D-друку під час процесу друку шляхом аналізу відео, знятого під час процесу.

Дефекти, ймовірно, виникнуть у 3D-друкованих об'єктах під час процесу друку, один із них – це струни; вони здебільшого пов'язані з одним із параметрів друку або геометрією об'єкта.

Нанизування дефектів може бути великого масштабу і зазвичай локалізується у видимих частинах об'єкта камерою захоплення.

У цьому випадку модель штучного інтелекту (Deep Convolutional Neural Network) була навчена на зображеннях, де чітко відображена проблема струнування, і розгорнуто в живому середовищі для виявлення та прогнозування на каналі відеокамери.

У цій роботі ми представляємо методологію розробки та розгортання глибоких нейронних мереж для розпізнавання рядків. Навчена модель може бути успішно розгорнута (з відповідним складанням необхідного обладнання, такого як

мікропроцесори та камера) у живому середовищі. Нанизування можна розпізнати відповідно до високої швидкості та точності класифікації.

Крім того, цей підхід може бути вдосконалений, щоб внести корективи в процес друку. Завдяки цьому запропонований підхід може або припинити процес друку, або виправити параметри, пов'язані з виявленим дефектом.[1]

Тимчасово можливо використовувати сторонній проект для аналізу зображення на предмет дефектів друку, але це не відповідає темі роботи.

Серед проектів з використання машинного зору для пошуку дефектів друкованої моделі, слід звернути увагу на Printpal.

Їх рішення вже розроблене, і розповсюджується за підпискою, та вичислює процеси аналізу на сервері. В такому випадку немає легальної можливості використати їх способи навчання і саму модель штучного інтелекту. На жаль, проект не підходить з принципових причин, хоча задає планку якості.

Printpal, компанія з Чикаго, яка спеціалізується на машинному навчанні, розробила програмне забезпечення на основі штучного інтелекту для виявлення дефектів у 3D-друку під назвою PrintWatch. Ось ключові особливості цього програмного забезпечення:

Основні функції PrintWatch:

1. Використання камери: PrintWatch використовує камеру, яка спрямована на зону будівництва FFF 3D-принтера, для моніторингу друкованих робіт.

2. Виявлення дефектів у реальному часі: програмне забезпечення визначає, коли саме починає формуватися дефект, і здатне відстежувати дефекти під час процесу 3D-друку, приймаючи реактивні заходи за необхідності.

3. Автоматичне втручання: PrintWatch може перервати друк, вимкнути нагрів принтера та відправити повідомлення про статус користувачам. Це забезпечує запобігання носіння дефектного друку протягом годин, зберігаючи час і філамент, а також знижуючи ризик пошкодження обладнання або пожежі.

Технологічні особливості:

1. Комп'ютерний зір: PrintWatch має власну модель машинного навчання, яка аналізує відеопотік у реальному часі і може виявити дефекти різних розмірів, форм, кольорів, матеріалів, освітлення та налаштувань.

2. Відстеження дефектів: програма стежить за дефектом, щоб перевірити, чи він погіршується, і втручається тільки тоді, що допомагає запобігти хибно позитивним сигналам.

3. Система виявлення аномалій: ця система завжди працює в фоновому режимі та призначена для виявлення інших аномалій, які можуть виникати в 3D-принтері, наприклад, проблеми з вирівнюванням осі.

Використання та сумісність:

PrintWatch виконує всі обчислення, пов'язані з машинним навчанням, у хмарі, тому користувачам не потрібен комп'ютер з високими технічними характеристиками для використання програмного забезпечення. Воно сумісне з будь-якою системою, яка має інтернет-з'єднання, включаючи простий Raspberry Pi.

Штучний інтелект для аналізу зображень зазвичай використовує конволюційні нейронні мережі (CNN).

CNN – це спеціалізовані нейронні мережі для обробки даних, що мають відомий, сітчастий формат, наприклад, зображення.

Конволюційні нейронні мережі (CNN, Convolutional Neural Networks), є основною технологією у сфері аналізу зображень і комп'ютерного зору. Цей термін походить від математичної операції, відомої як конволюція (або згортка), яка є ключовою частиною архітектури цих мереж.

Ось основні моменти про конволюційні нейронні мережі:

Конволюція (згортка): конволюційні шари використовують фільтри (або ядра), які "проходять" по вхідному зображенню, щоб створити карти особливостей (feature maps).

Кожен фільтр автоматично виявляє специфічні особливості на різних рівнях зображення, такі як краї, кути, кольори або більш складні патерни.

Роль конволюційних шарів: ці шари допомагають CNN визначати спатіальні ієрархії особливостей в зображеннях, тобто вони дозволяють мережі "розуміти"

структуру та зміст зображення на більш глибокому рівні, від простих форм до складних об'єктів.

Навчання фільтрів: у традиційному машинному навчанні фільтри або особливості потрібно було визначати вручну. Однак, у конволюційних нейронних мережах ці фільтри вивчаються автоматично під час процесу тренування, дозволяючи мережі адаптуватися до конкретних особливостей даних.

Архітектура CNN: крім конволюційних шарів, CNN також включають пулінгові шари (що зменшують розмір карти особливостей, зберігаючи при цьому важливу інформацію) і повнозв'язні шари (що використовуються для класифікації або регресії на основі виявлених особливостей).

Конволюційні нейронні мережі стали фундаментом для багатьох проривів у комп'ютерному зорі та аналізі зображень, включаючи розпізнавання облич, автоматичний аналіз медичних зображень, системи відеонагляду, і навіть у додатках штучного інтелекту в автомобільній промисловості.

EfficientNet Lite - це оптимізована версія моделі штучного інтелекту EfficientNet, розроблена для працездатності на пристроях із обмеженими ресурсами, наприклад, на Raspberry Pi. EfficientNet, оригінальна модель, була створена дослідниками з Google AI і представлена у 2019 році. Основною ідеєю цієї моделі було збалансоване масштабування всіх трьох вимірів мережі: ширини, глибини та розміру зображення.

Основні дані про EfficientNet Lite:

1. Оптимізація: оптимізована спеціально для Мобільних та Edge-пристроїв. EfficientNet Lite була спеціально адаптована для працездатності на пристроях із обмеженим обчислювальним потенціалом, таких як мобільні телефони та Raspberry Pi. Це досягається за рахунок спрощення архітектури та зниження вимог до обчислювальних ресурсів.

2. Використання: EfficientNet Lite широко використовується в задачах комп'ютерного зору, таких як класифікація зображень, виявлення об'єктів, аналіз відео тощо. Вона особливо корисна в сценаріях, де необхідна висока точність при обмежених ресурсах.

3. Медична діагностика: EfficientNet була використана для аналізу медичних зображень, наприклад, для виявлення раку шкіри, аналізу рентгенівських знімків, мамограм і т.д. Завдяки своїй високій точності та здатності розпізнавати складні патерни на зображеннях, вона допомагає лікарям у постановці більш точних діагнозів.

4. Автономне водіння: у сфері автономного водіння EfficientNet використовується для обробки і інтерпретації зображень з камер, що дозволяє автомобілям краще орієнтуватися в просторі, розпізнавати дорожні знаки, перешкоди, інші транспортні засоби та пішоходів.

5. Екологічний моніторинг: модель також знайшла застосування у проектах, пов'язаних з моніторингом довкілля, наприклад, для аналізу супутникових зображень для виявлення змін у ландшафті, відслідковування вирубки лісів або змін водних ресурсів.

5. Розпізнавання об'єктів і людей: в секторі безпеки та відеонагляду EfficientNet використовується для ідентифікації об'єктів або людей на відео, що може допомогти у питаннях безпеки та моніторингу.

6. Роздрібна торгівля та реклама: в роздрібній торгівлі та рекламі EfficientNet використовується для аналізу зображень продуктів, що може допомогати у класифікації товарів, аналізі трендів та навіть у персоналізованому маркетингу.

7. Агро технології: у сільському господарстві модель може застосовуватися для аналізу зображень рослин для виявлення захворювань рослин або шкідників, що допомагає вчасно вжити заходів для захисту врожаю.

Таблиця 2.2

Порівняння моделей

Назва	Особливості	Переваги	Недоліки
Mobile Net	Спеціалізується на мінімізації обчислювального навантаження. Вона використовує так звані глибокі роздільні згортки для зменшення кількості параметрів та операцій.	Є дуже ефективною для мобільних пристроїв та інших ситуацій з обмеженими обчислювальними ресурсами.	Може відставати від більш потужних моделей, таких як efficientnet, з точки зору точності у складних завданнях комп'ютерного зору.
Shuffle Net	Використовує унікальну техніку змішування каналів для підвищення ефективності згорткових операцій.	Є надзвичайно легкою та ефективною з точки зору обчислювальних витрат, що робить її ідеальною для пристроїв з обмеженими ресурсами.	Подібно до mobilenet, може поступатися у точності більш складним моделям.
Squeeze Net	Відома своєю екстремальною ефективністю у використанні параметрів, використовуючи "fire modules" - спеціальні блоки, які зменшують кількість параметрів.	А є дуже маленький розмір, що робить її підходящою для вбудованих систем та мобільних додатків.	Маючи дуже малий розмір, вона може не досягати такої ж точності, як більш великі та складні моделі.

Порівняння з EfficientNet

EfficientNet забезпечує кращий баланс між ефективністю та точністю. Завдяки своєму унікальному методу "Compound Scaling", вона забезпечує високу точність навіть на складних завданнях, зберігаючи при цьому обчислювальну ефективність.

EfficientNet пропонує більшу гнучкість у налаштуванні, що дозволяє їй адаптуватися до різних обчислювальних потреб і завдань.

Хоча всі ці моделі широко використовуються в області комп'ютерного зору, EfficientNet знайшла більш широке застосування у більш складних сценаріях завдяки своїй високій точності та ефективності.

Методи навчання машинного зору базуються на різних підходах до машинного навчання, кожен з яких має свої унікальні особливості та застосування. Основні методи включають навчання з вчителем, без вчителя, напівнаглядове навчання, посилене навчання, перенос навчання, глибоке навчання, активне навчання, та мета-навчання. Кожен з цих методів використовує різні стратегії для аналізу та інтерпретації візуальних даних, виходячи з конкретних потреб та наявних ресурсів.

Перелік методів:

1. Навчання з вчителем (Supervised Learning): цей метод передбачає використання попередньо маркованих датасетів, де кожному вхідному зображенню відповідає мітка чи опис. Алгоритм намагається вивчити взаємозв'язки між зображеннями та їх мітками, щоб потім можна було класифікувати нові дані.

2. Навчання без вчителя (Unsupervised Learning): тут алгоритми аналізують неанотовані датасети, намагаючись виявити приховані шаблони або структури без попереднього навчання на маркованих даних. Це може включати кластеризацію зображень за певними ознаками.

3. Напівнаглядове навчання (Semi-supervised Learning): комбінує марковані та немарковані дані під час навчання. Це дозволяє системі краще узагальнювати інформацію з обмеженої кількості маркованих даних.

4. Посилене навчання (Reinforcement Learning): моделі навчаються за допомогою системи винагороди, де правильні дії збільшують бали або «винагороду», а неправильні - зменшують. Це часто використовується у робототехніці та іграх.

5. Перенос навчання (Transfer Learning): використовує моделі, які були навчені на одному наборі даних, для вирішення схожих завдань на іншому наборі даних. Це ефективно, коли для нової задачі обмежена кількість даних.

6. Глибоке навчання (Deep Learning): підхід, який використовує нейронні мережі з багатьма шарами (глибокі мережі) для виявлення складних шаблонів у великих обсягах даних. Це особливо ефективно для розпізнавання зображень та відео.

7. Активне навчання (Active Learning): метод, де модель вибирає, які дані їй потрібно анотувати та використовувати для навчання, щоб покращити свою продуктивність.

8. Мета-навчання (Meta-Learning): також відоме як "навчання навчання", це підхід, де моделі навчаються адаптуватися до нових завдань швидко з мінімальним набором даних.

Навчання з вчителем можна класифікувати як надійне навчання, дані були попередньо помічені по критеріям людиною, таким чином можна, наприклад, помітити дефекти поверхні і дати ці дані інтелекту на навчання, і вона не зможе хибно навчатися, тільки страждає ефективність процесу через повільне маркування яке сповільнене людськими можливостями.

Інший спосіб притаманний для машинного зору – це глибоке навчання. Машинний зір це глибокі мережі які використовують складні алгоритми з більшою аналізуючою можливістю, в них масштабовані фільтри, шари, тому їм легше навчатись самостійно під наглядом.

2.3. Процес навчання методом вчителя

Першочергово потрібен алгоритм дій для навчання (рис. 2.6).

Перший етап – збір матеріалів навчання.

Перед тим як почати навчання необхідні дані за якими буде навчатися нейронна мережа, для розпізнавання конкретних елементів поверхні чи об'єктів.

У звичайних випадках навчання машинного зору для ідентифікування об'єктів використовують підготовлені бібліотеки зображень, у певному форматі. Наприклад, зображення 28 на 28 пікселів у чорно білому стилі. Такий спосіб обробки зображень необхідний для великих бібліотек і оптимізації часу навчання, неймережі простіше аналізувати тисячі зображень, якщо вони малі, таким чином вона навчиться класифікуванню об'єктів. Даний спосіб є застарілим, неймережі здатні швидко обробляти великі зображення у пошуку потрібних об'єктів, для ідентифікації використовується більша кількість шарів обробки зображення, а потужності систем на яких нейронні мережі працюють і навчаються вироста.

Слід використовувати будь які зображення, серед них:

- зроблені власноруч;
- запозичені з інтернет джерел;
- створенні у редакторах;
- згенеровані.

Перш за все потрібні зображення ярко виражених дефектів, походження не важливо, вони можуть бути намальовані іншою неймережею, навчання все одно буде ефективним.

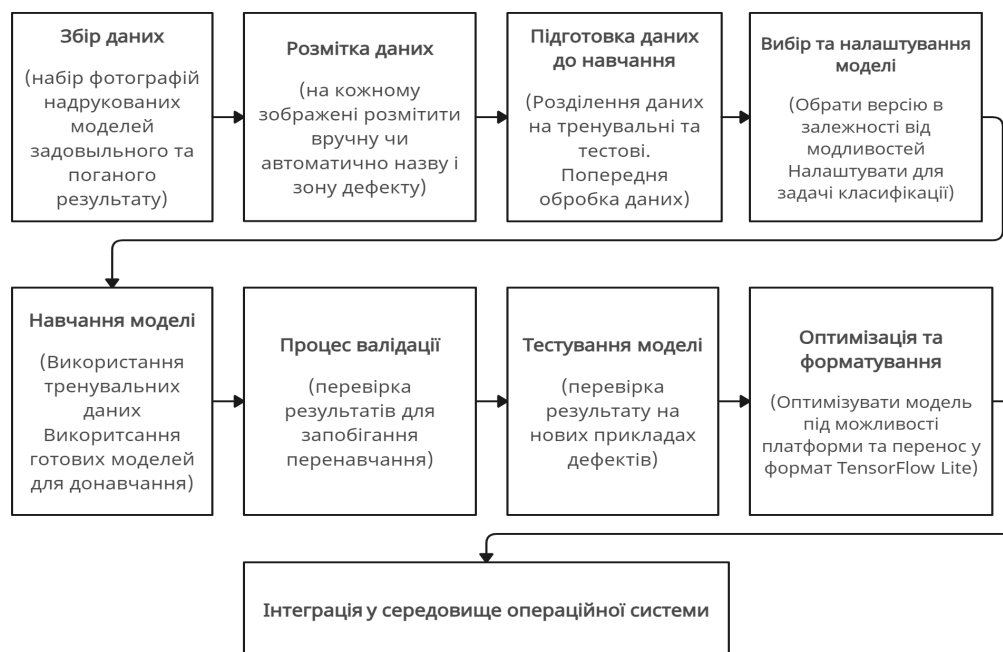


Рис.2.6. Алгоритм навчання методом вчителя

Розмітка матеріалів навчання.

Після забору матеріалу для навчання необхідно їх підготувати, помітивши зони, які машинний зір буде аналізувати, після того як пропустить ці зображення через фільтри глибокого навчання, знайде рівні і карти, на яких буде найбільш виражені аспекти, після цього нейромережа стане шукати подібні збіги в майбутньому.

Сама розмітка виконується через спеціальні програми та сервіси.

Такі як:

- **V7 Labs:** цей інструмент підтримує автоматизовані анотації, робочі процеси, що складаються, керування наборами даних, співпрацю в реальному часі та підходить для анотацій медичних зображень. Він універсальний для унікальних типів файлів, включаючи формати надвисокої роздільної здатності та мультиспектральні формати.
- **Labelbox:** відомий своїм маркуванням за допомогою штучного інтелекту та настроюваним інтерфейсом, Labelbox підтримує анотації з багатокутниками, обмежувальними рамками та лініями. Він чудово підходить для таких завдань, як розпізнавання об'єктів і семантична сегментація.
- **Scale AI:** цей інструмент добре підходить для анотацій великих обсягів 3D-датчиків, зображень і відеоданих, особливо корисних для додатків автономного водіння. Він оснащений системами попереднього маркування та автоматизованими системами контролю якості на базі ML.
- **SuperAnnotate:** розроблена як для анотацій зображень, так і для відео, ця платформа пропонує інструменти для векторних і піксельних анотацій. Він підтримує низку завдань комп'ютерного зору та забезпечує розширені системи контролю якості.
- **Dataloop:** універсальна хмарна платформа анотацій, Dataloop підтримує як зображення, так і відеодані та орієнтована на весь життєвий цикл штучного інтелекту, включаючи оцінку та вдосконалення моделі.

- Markup Hero: цей інструмент дозволяє користувачам коментувати фотографії, підкреслюючи важливу інформацію, додаючи мітки, обрізаючи, змінюючи розмір і обертаючи зображення. Він зручний і підходить для взаємодії та спілкування в реальному часі.
- Keylabs: пропонує низку стилів анотацій, включаючи обмежувальні рамки та семантичну сегментацію, із вбудованими елементами контролю якості та підходами до розширення даних.
- CVAT (Інструмент анотації комп'ютерного зору): цей інструмент із відкритим кодом, розроблений компанією Intel, підтримує виявлення об'єктів, класифікацію та сегментацію зображень. Він пропонує напівавтоматичне анотування та низку інструментів автоматизації.
- Labellmg: інструмент на основі Python, простий у використанні для наборів даних до 10 000 зображень, чудово підходить для моделей виявлення об'єктів.
- Viso Suite: ця платформа містить вбудоване середовище для анотацій зображень і підтримує спільну роботу, дозволяючи користувачам збирати відеодані, додавати анотації до зображень, навчати та керувати моделями AI

Для роботи був обраний саме CVAT, який відповідав параметрам даної роботи.

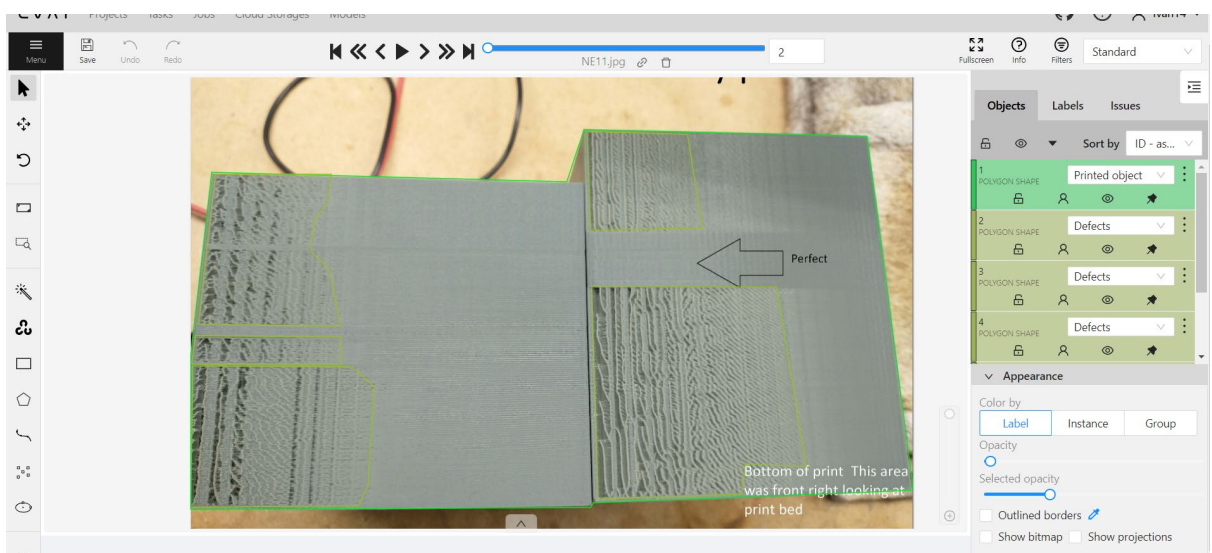


Рис. 2.7. Розмітка об'єкту друку та дефектів недостатньої екструзії.

В даному випадку виникає ряд труднощів.

Всі фотографії повинні бути однієї роздільної здатності, наприклад якість 1920 на 1080 Full HD можлива, але витрачає більше ресурсів завдяки якості зображення, яке здатне також покращити можливості пошуку нейромережею.

Зображення повинні бути чіткими та без значних спотворень або шумів. Погана якість може негативно вплинути на точність моделі.

Датасет повинен включати різноманітність образів, сцен, освітлення та перспектив, щоб модель могла узагальнювати своє вивчення на різні умови.

Анотації мають бути точними та послідовними. Неправильні або неточні анотації можуть призвести до помилкового навчання моделі.

Якщо модель використовується для виявлення об'єктів, розміри та пропорції об'єктів у зображеннях повинні бути зрозумілими і повними, щоб модель могла ефективно їх розпізнавати.

Варіації у кольорі та освітленні можуть впливати на виконання моделі, тому важливо включати зображення з різними умовами освітлення та кольоровою гамою.

Зображення мають бути у форматі, який підтримується інструментарієм для тренування моделі (наприклад, JPEG, PNG).

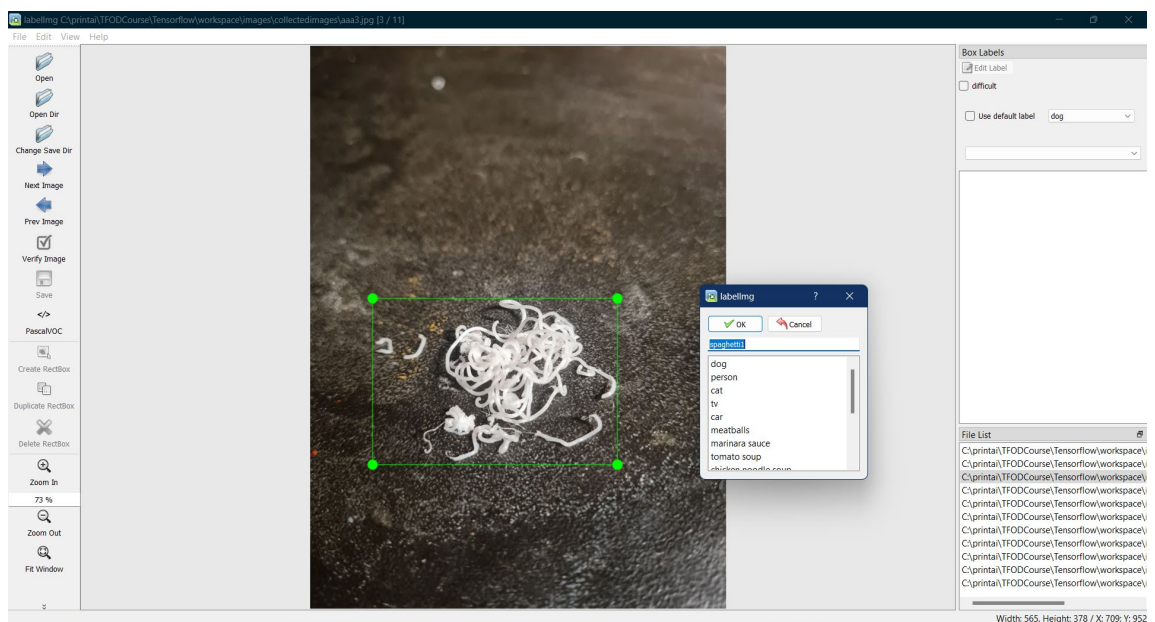


Рис. 2.8. Використання внутрішньої програми розмітки.

Продемонстрована вище програма також здатна розмічати дані для навчання, за відмінністю того, що вона встановлена за допомогою командної строки під час створення проекту по навчанню машинного зору на бібліотеці TensorFlow.

Сама підготовка даних для навчання вже є складною, критичні умови як роздільна здатність повинні бути в пріоритеті, інакше помилки і неможливість навчання гарантовані.

Підготовка до навчання.

Для роботи з нейромережами використовують мову програмування Python.

Python є популярною мовою програмування для роботи з нейронними мережами у сфері машинного зору з кількох причин:

Має широкий спектр бібліотек для машинного навчання та глибокого навчання, таких як TensorFlow, PyTorch, Keras, OpenCV, що спрощують розробку та імплементацію нейронних мереж.

Відомий своєю простотою синтаксису та читабельністю коду, що робить його доступним навіть для тих, хто тільки починає роботу з програмуванням.

Існує велика та активна спільнота розробників Python, яка постійно розробляє нові інструменти та надає підтримку через форуми та документацію.

Python легко інтегрувати з іншими системами та інструментами для обробки даних, що важливо у великих проектах машинного зору.

Наукові обчислення: бібліотеки Python, такі як NumPy та SciPy, забезпечують потужні можливості для наукових обчислень, які є основою для обробки даних та навчання нейронних мереж.

Дозволяє легко експериментувати з різними архітектурами нейронних мереж, а також розширювати можливості за допомогою власних розробок або інтеграції сторонніх бібліотек.

Python — це крос-платформна мова, що дозволяє розробникам легко розгорнути програми на різних операційних системах.

Завдяки цим перевагам, Python став стандартом де-факто для багатьох досліджень та проектів у сфері машинного зору та глибокого навчання.

Встановити Python треба з офіційного сайту, після цього необхідне середовище для проекту навчання машинного зору.

Необхідно встановити програмний засіб “Anaconda”.

Anaconda — це популярний дистрибутив Python, який використовується в основному для наукових обчислень, аналізу даних, інженерії даних, машинного навчання та інших областей, де використовуються великі обсяги даних та складні наукові обчислення. Ось основні особливості та переваги Anaconda:

1. Управління пакетами та середовищами.

Anaconda використовує інструмент Conda для управління пакетами та середовищами. Conda дозволяє легко встановлювати, запускати та оновлювати пакети, а також створювати та керувати ізольованими середовищами. Це особливо корисно, коли одночасно працюють декілька проектів, які вимагають різних версій бібліотек або навіть Python.

2. Велика кількість пакетів.

Anaconda включає в себе сотні пакетів для наукових обчислень, аналізу даних, машинного навчання та візуалізації. Багато з цих пакетів вже попередньо встановлені в Anaconda, що робить її швидким і зручним рішенням для розгортання наукових та аналітичних проектів.

3. Легкість використання.

Anaconda має графічний інтерфейс користувача, Anaconda Navigator, який робить роботу з Conda та управління середовищами зручнішою та інтуїтивною.

4. Підтримка різних мов.

Хоча Anaconda зосереджена на Python, вона також підтримує інші мови, такі як R — ще одна популярна мова для аналізу даних та статистики.

5. Використання в академічних та промислових сферах.

Anaconda широко використовується університетами, дослідниками, а також у промисловості для розвитку складних наукових та аналітичних додатків.

6. Інтеграція з інструментами розробки.

Anaconda легко інтегрується з популярними інструментами та сервісами розробки, такими як Jupyter Notebook, Spyder (IDE для наукових обчислень у Python), Microsoft VSCode тощо.

7. Вільне та відкрите ПЗ

```
Microsoft Windows [Version 10.0.22635.2486]
(c) Корпорація Майкрософт (Microsoft Corporation). Все права захищені.

C:\printai>git clone https://github.com/nicknochnack/TFODCourse
Cloning into 'TFODCourse'...
remote: Enumerating objects: 89, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 89 (delta 23), reused 21 (delta 21), pack-reused 58
Receiving objects: 96% (86/89), 29.82 MiB | 478.00 KiB/s
Receiving objects: 100% (89/89), 30.02 MiB | 451.00 KiB/s, done.
Resolving deltas: 100% (35/35), done.

C:\printai>-m venv diplo
"-m" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\printai>cd TFODCourse

C:\printai\TFODCourse>python -m venv diplo

C:\printai\TFODCourse>.\diplo\scripts\activate

(diplo) C:\printai\TFODCourse>python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\printai\TFODCourse\diplo\lib\site-packages (23.0.1)
Collecting pip
  Obtaining dependency information for pip from https://files.pythonhosted.org/packages/bd199be08cdb4443a02e26bd2c3cf6fc39/pip-23.3.2-py3-none-any.whl.metadata
  Downloading pip-23.3.2-py3-none-any.whl.metadata (3.5 kB)
  Downloading pip-23.3.2-py3-none-any.whl (2.1 MB)
----- 2.1/2.1 MB 1.7 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 23.2.1
    Uninstalling pip-23.2.1:
      Successfully uninstalled pip-23.2.1
  Successfully installed pip-23.3.2

(diplo) C:\printai\TFODCourse>pip install ipykernel_
```

Рис. 2.9. Процес створення середовища для нейромережі

Anaconda є безкоштовним та відкритим програмним забезпеченням, хоча існує комерційна версія Anaconda Enterprise, яка надає додаткові функції та підтримку для корпоративних клієнтів.

Використання Anaconda дозволяє швидко налаштувати середовище для роботи з даними та науковими обчисленнями, мінімізуючи необхідність вручну вирішувати залежності та конфлікти між пакетами.

3. РОЗРОБКА ПРОГРАМИ ЗАБЕЗПЕЧЕННЯ ТА НАЛАШТУВАННЯ РОБОТИ 3D ПРИНТЕРА

3.1. Робота з веб-інтерфейсом

Для роботи з принтером через веб-інтерфейс необхідно його створити і модифікувати.

Після встановлення операційної системи OctoPi на міні-комп'ютер Raspberry, йде налаштування інтерфейсу доступного через веб-додаток "OctoPrint". Стандартний простий інтерфейс, можна переробити для зручності і візуального задоволення. Також не всі функції працюють одразу, їх необхідно переналаштувати, що можна зробити під час переробки інтерфейсу з нуля створивши його варіанти, включно з варіантами для формату мобільного телефону.

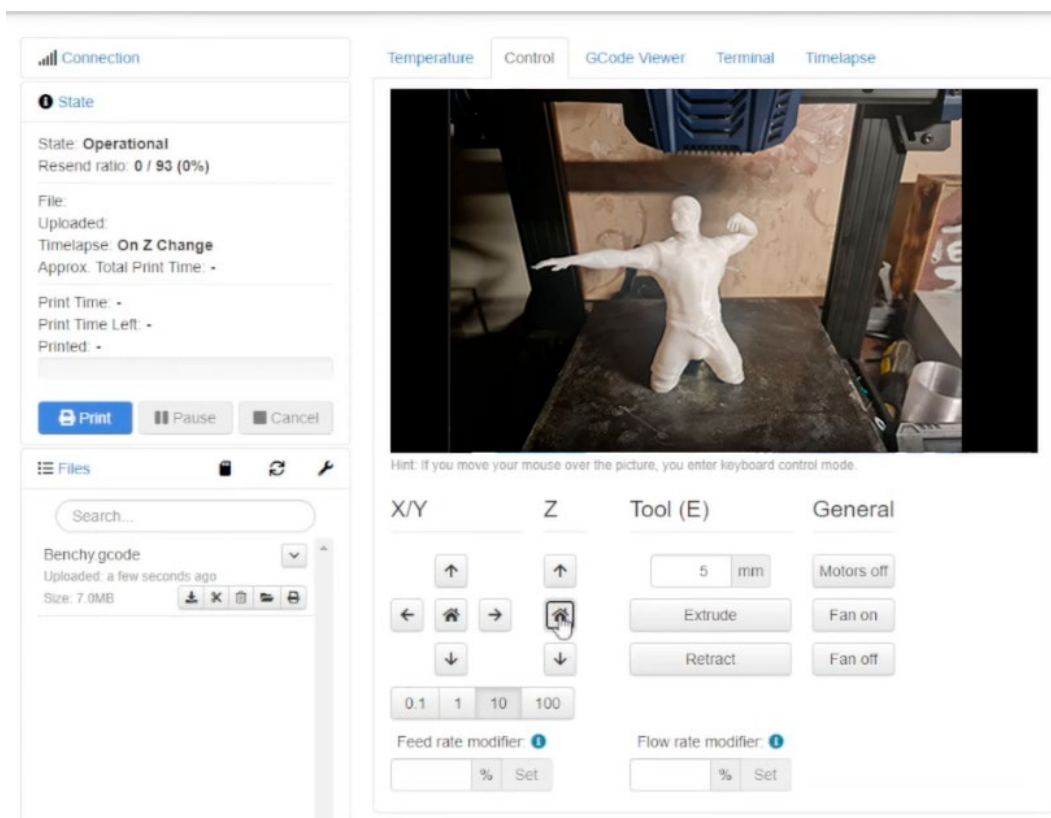


Рис. 3.10. Інтерфейс віддаленого керування

Завдяки відкритому джерелу, проект має вбудовані теми та плагіни для тонкого налаштування, також користувачі розробляють власні рішення та плагіни, які здатні розширювати функціонал і поєднувати роботу додаткових програм.

Після встановлення операційної системи OctoPi на міні-комп'ютер Raspberry Pi користувач запускає веб-додаток OctoPrint для подальшого налаштування через веб-інтерфейс.

Відкривши веб-браузер на пристрої, підключеному до мережі, користувач вводить IP-адресу Raspberry Pi, де запущено OctoPrint. Це дає змогу отримати доступ до інтерфейсу управління 3D принтером.

Після входу в інтерфейс OctoPrint користувач налаштовує параметри мережі, встановлює пароль доступу для забезпечення безпеки та обирає налаштування для підключення до конкретного принтера.

Далі, за допомогою USB-кабелю, принтер фізично підключається до Raspberry Pi. У веб-інтерфейсі OctoPrint користувач обирає свій принтер та проводить його налаштування, включаючи встановлення параметрів друку.

Для повноцінної роботи веб-інтерфейсу необхідно розробити власний дизайн перевіряючи кожну кнопку та сторінку інтерфейсу.

Необхідність насправді гостра, доступний інтерфейс малоінформативний і викликає складнощі у пошуку даних як вручну так і візуально, слід починати дизайн веб-інтерфейсу з власної головної сторінки, яка буде включена за замовчуванням.

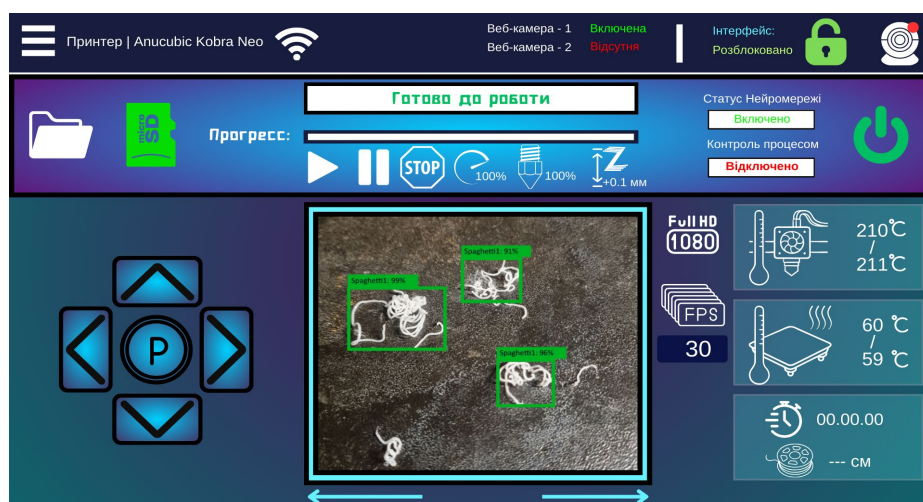


Рис. 3.11. Власний головний інтерфейс

Його створення займає час, через необхідність назначати кожному елементу команду, анімацію для розуміння віддачі інтерфейсу. Також зберігається можливість подальшого розвитку та повної переробки. Сам код інтерфейсу повністю складно представити, через частковий допуск до коду, та закодованість повного коду.

Перша, головна сторінка інтерфейсу містить всю необхідну інформацію, знаходиться в розробці, тому кількість інформації та стиль інтерфейсу буде змінюватись.

OctoPrint дозволяє створювати профілі друку з різними параметрами, такими як швидкість друку, температура і тип матеріалу.

Користувач отримує змогу відстежувати процес друку, контролювати його (пауза, відновлення, скасування) та взаємодіяти з принтером віддалено через веб-інтерфейс.

Для розширення функціональності користувач може встановлювати додаткові плагіни, які розширюють можливості OctoPrint, забезпечуючи нові функції, такі як моніторинг відео, сповіщення тощо.

Після завершення налаштувань, рекомендується здійснити тестовий друк для перевірки налаштувань та встановлення оптимальних параметрів.

Для забезпечення безпеки й оптимальної роботи пристрою, користувач повинен регулярно оновлювати OctoPrint та використовувати належні налаштування доступу та паролі.

Цей алгоритм дозволяє налаштувати та користуватися OctoPrint для управління 3D принтером через веб-інтерфейс, надаючи зручний спосіб керування процесом друку.

3.2 Додатковий мобільний додаток

Подібний інтерфейс до інтерфейсу OctoPrint доступний у вигляді мобільного додатку “**Printoid**”, який отримує дані від вбудованої в операційну систему програми OctoPrint, даючи можливість використовувати більш зручний інтерфейс

для мобільних пристроїв, а завдяки відкритому коду інтерфейс можна налаштувати під себе.

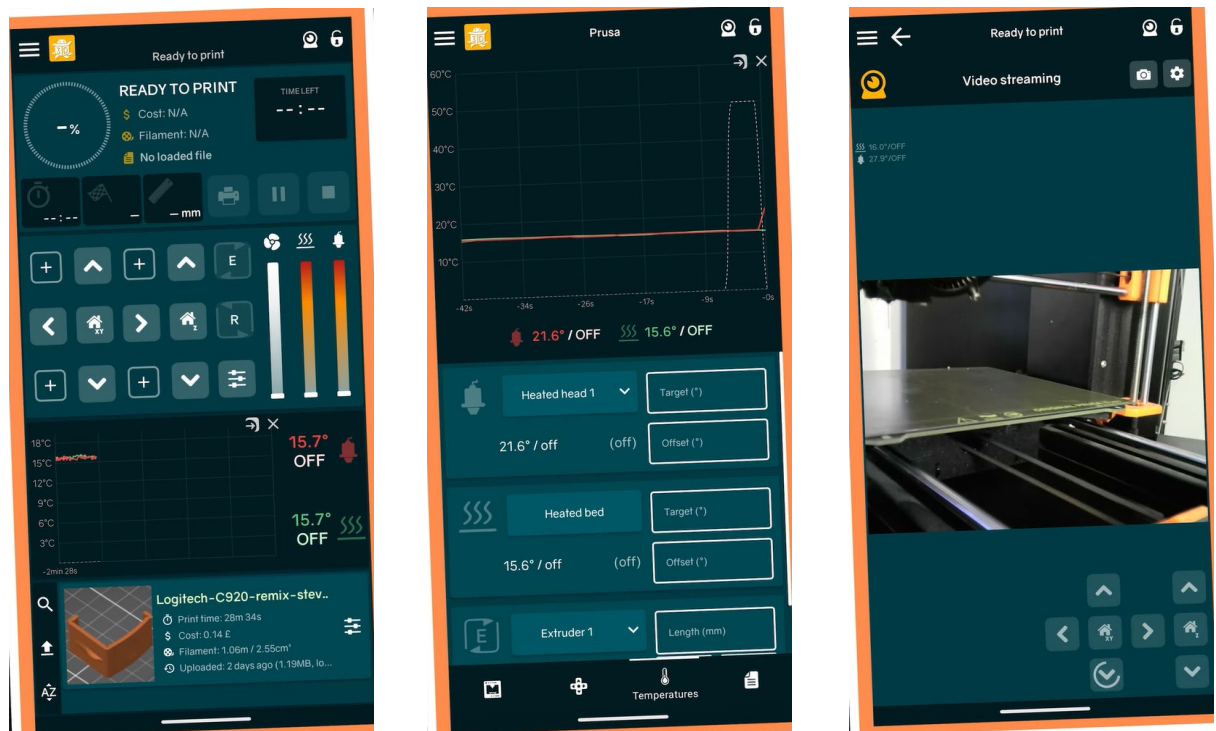


Рис. 3.12. Мобільний додаток “Printoid”

Printoid є додатком для мобільних пристроїв на платформі Android, який надає користувачам можливість віддалено керувати та моніторити їх 3D принтери, що працюють з відкритим програмним забезпеченням OctoPrint. OctoPrint надає веб-базований інтерфейс для керування 3D принтерами через браузер, тоді як Printoid пропонує інтерфейс, оптимізований для мобільних пристроїв, з додатковими функціями, які роблять керування принтером зручним прямо з смартфона чи планшета.

Дана програма розповсюджується відкрито, тому її легко модифікувати, чи використовувати частини коду для власних потреб без проблем з правами власності.

Було прийнято рішення розробити власну легку версію мобільного додатку для керування принтером.

Printoid дозволяє користувачам виконувати багато тих же дій, що й OctoPrint, включаючи запуск друку, паузу або зупинку друкованих завдань, перегляд стану друку в реальному часі, контроль температур, перегляд відео в реальному часі з камери, встановленої на принтері, та отримання сповіщень. Тобто, програма дублює можливості контролю, і має власний інтерфейс, також присутні певні додаткові функції, які рідко використовуються.

Для індивідуальності і простоти проекту, була почата розробка власного програмного засобу для смартфона, з власним дизайном який буде мати варіанти під настрій, з простими можливостями налаштування кольорів і налаштуванням видимості частин інтерфейсу. Головне, це спрощення функціоналу.

Після певних труднощів з обранням середовища розробки, яке не буде зазначатись в роботі, через міграцію проекту по різних програмах, внаслідок певних складнощів і малого знання у даній темі, було створено прототип власної програми яка частково базована на відкритому коді іншої програми "Printoid".

Як результат, присутні перші основні сторінки, які дублюють дизайн веб-інтерфейсу для браузера, але частини розбиті по сторінках, через незручність малого екрану телефону і складності у натисканні потрібного елемента.

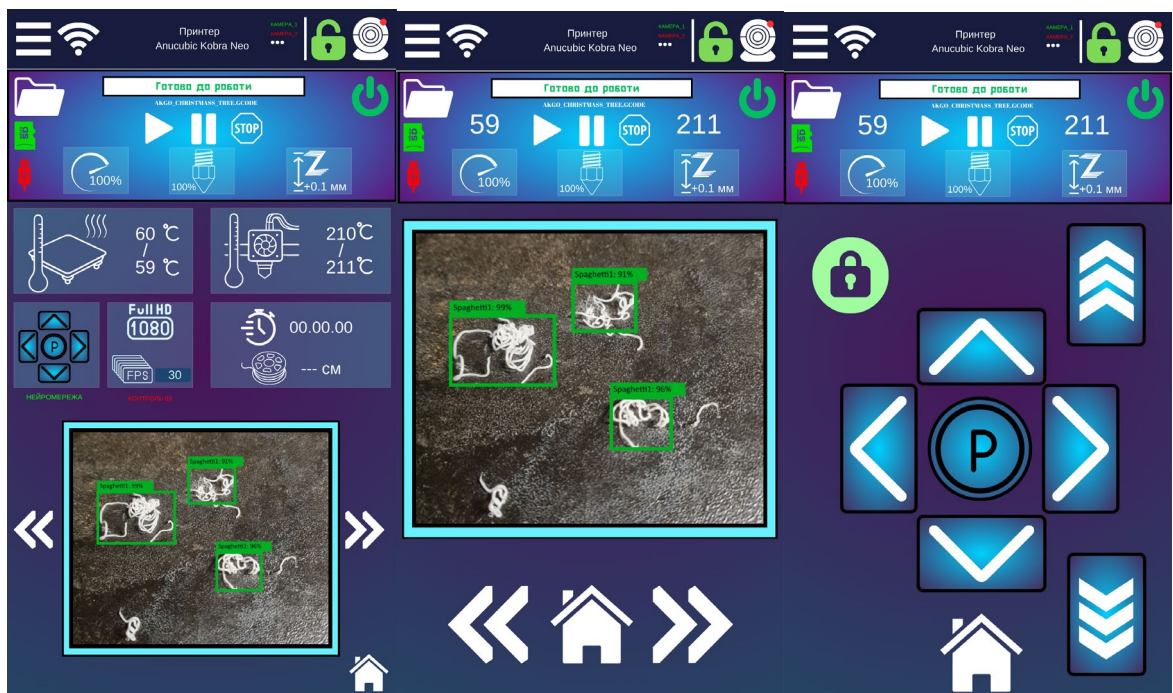


Рис. 3.13. Власний мобільний додаток "Prototip"

Функціональність додатку має всі стандартні можливості OctoPrint, без додаткових можливостей, за виключенням додавання вікна камери, на якій відслідковується роботи нейромережі, яка використовує комп'ютерний зір для пошуку дефектів друку.

Основні функції додатку:

1. Віддалене керування принтером: користувач може віддалено керувати процесом друку через мобільний пристрій, такі як: пауза, відновлення чи скасування друку, процент подачі пластику, налаштування температур та обдуву, а також керування положенням друкуючої голови чи поверхні друку і вільний від процесу друку час.
2. Моніторинг процесу друку: додаток дає змогу спостерігати за процесом друку в реальному часі через веб камеру, мобільний телефон або планшет. Присутня можливість переглядати статус друку, температуру принтера, швидкість та інші параметри.
3. Реалізована у тестовому стані функція, яка дозволяє користувачеві керувати файлами, що готуються до друку, надсилаючи, переглядаючи та видаляючи їх.
4. Гнучкість інтерфейсу: завдяки відкритому коду інтерфейсу, можливо налаштувати додаток під свої потреби, включаючи розміщення, видимість та вигляд елементів управління.
5. Оповіщення та інформація: присутні сповіщення для користувача про статус друку або про потребу уваги до принтера.

Цей додаток забезпечує більш зручний та мобільний спосіб керування процесом друку 3D моделей через OctoPrint, дозволяючи користувачеві контролювати та моніторити друкування навіть з віддаленої локації за допомогою свого мобільного пристрою.

Алгоритм роботи мобільного додатку "Prototip" для керування 3D принтером через OctoPrint можна узагальнити наступним чином:

1. Підключення до OctoPrint: користувач встановлює додаток на свій мобільний пристрій та запускає його.

2. Налаштування підключення: у додатку відбувається налаштування з'єднання з сервером OctoPrint, вводиться IP-адреса Raspberry Pi або URL для підключення.
3. Аутентифікація: якщо вимагається аутентифікація, користувач вводить логін та пароль для доступу до сервера OctoPrint, але відключено для легкості доступу під час тестування.
4. Отримання даних з OctoPrint: після успішного підключення, додаток отримує дані про статус принтера, активні процеси друку, температуру, файли для друку та іншу інформацію.
5. Відображення інтерфейсу: відображається інтерфейс, який надає користувачеві зручну навігацію та управління. Це може бути список файлів, статус друку, налаштування контролю тощо.
6. Управління процесом друку: користувач може керувати друкуванням: запускати, призупиняти, відновлювати або скасовувати друк.
7. Перегляд статусу друку: додаток відображає актуальну інформацію про хід друку: відсоток завершення, час залишку, температуру, можливі помилки тощо.
8. Керування файлами: користувач може переглядати, надсилати, видаляти або редагувати файли для друку через інтерфейс додатку.
9. Сповіщення та налаштування: може надсилати сповіщення користувачеві про важливі події під час друку. Крім того, додаток дозволяє користувачеві налаштовувати різні параметри, включаючи вигляд інтерфейсу та функціональність.
10. Завершення роботи: після закінчення процесу друку або при від'єднанні від сервера OctoPrint, додаток може завершити роботу або залишити можливість подальшого використання в офлайн для перегляду інформації.

3.3. Програмні рішення

Також є можливість написати код який буде реагувати на повідомлення від нейромережі при ідентифіковані дефекти, скрипт.

Коли дефект знайдений, скрипт відправляє HTTP-запит до OctoPrint для паузи друку.

Використати REST API для відправки команди на паузу.

Наприклад, через requests у Python:

```
import requests
OCTOPRINT_API_KEY = 'api_key_number_a1'
headers = {
    'Authorization': 'Bearer {}'.format(OCTOPRINT_API_KEY),
    'Content-Type': 'application/json'
}
response = requests.post('http://octoprint.local/api/job', headers=headers,
json={'command': 'pause', 'action': 'pause'})
```

Після цього у програмі Printoid створити автоматичне сповіщення при паузі друку, в такому випадку користувач сам вирішить повністю зупинити друк чи продовжити.

Крім реагування на дефекти та автоматичної паузи друку, можна розглянути інші програмні рішення для поліпшення управління процесом друку 3D:

1. Автоматичне відновлення друку: після усунення дефекту або проблеми можна налаштувати автоматичне відновлення друку. Додаток може надсилати команди до OctoPrint для продовження друку після паузи.
2. Системи сповіщень: розробка системи сповіщень, що дозволяє відстежувати та повідомляти користувача про статус друку, виявлені проблеми чи паузу.
3. Підтримка користувача через додаток: додаток може мати функцію підтримки користувачів, де можна отримати поради щодо виявлених дефектів та можливих дій для їх усунення.
4. Віддалене керування параметрами друку: розширення можливостей додатку для зміни параметрів друку (температури, швидкості тощо) в реальному часі через мобільний додаток.

5. Моніторинг та аналіз даних: збір та аналіз даних про друкування для виявлення паттернів, удосконалення якості друку та уникнення майбутніх проблем.
6. Інтеграція з хмарними сервісами: можливість зберігання даних про друк у хмарному сховищі для резервного копіювання та доступу до них з різних пристроїв.

Ці програмні рішення можуть поєднувати в собі функції для поліпшення управління процесом друку, забезпечуючи зручний та ефективний спосіб керування 3D принтером.

3.4 Підготовка та навчання машинного зору

Певні труднощі перед роботою з нейромережею полягають у виконанні обов'язкових вимог.

Підготовка середовища для Windows.

Необхідно встановити Python перейшовши на офіційний сайт Python (python.org) та завантажити останню версію Python (3.6+). Встановлюючи програму обов'язково вибрати опцію "Add Python to PATH".

Далі треба встановити бібліотеки через PIP відкривши командний рядок, натиснути Win + R, і ввести cmd і натиснути Enter.

Щоб встановити TensorFlow в командному рядку ввести: "pip install tensorflow". TensorFlow автоматично встановить всі необхідні залежності.

Встановити додаткові бібліотеки для аналізу даних та обробки зображень, також можуть знадобитися додаткові бібліотеки, наприклад: NumPy і Pandas: "pip install numpy pandas".

Налаштування віртуального середовища (опціонально):

1. Створення віртуального середовища:

В командному рядку ввести: "python -m venv myenv".

myenv - це назва віртуального середовища.

2. Активація віртуального середовища:

На Windows введіть: `myenv\Scripts\activate`

З'явиться назва віртуального середовища в командному рядку, якщо воно активовано.

3. Перевірка інсталяції:

Перевірка TensorFlow, в командному рядку ввести:

```
“python -c "import tensorflow as tf; print(tf.__version__)"
```

Це повинно вивести версію TensorFlow, що підтверджує успішну інсталяцію.

Також може знадобитися встановити додаткове програмне забезпечення, наприклад, Git для контролю версій або Jupyter Notebook для інтерактивної розробки, теж командно.

Для деяких операцій можуть знадобитися права адміністратора.

3.5 Підготовка даних і розмітка даних

На вибір доступні два інструменти: LabelImg та CVAT.

LabelImg: простий у використанні інструмент для розмітки зображень, що працює локально. Підходить для невеликих проєктів або коли потрібен швидкий старт. Підтримує формати розмітки, такі як PASCAL VOC (XML) та YOLO (текстові файли).

CVAT (Computer Vision Annotation Tool): веб-базований інструмент, більш потужний та гнучкий у порівнянні з LabelImg. Підходить для більших проєктів з командною роботою.

Підтримує різноманітні формати розмітки та анотації, включаючи об'єкти, полігони, кеглі тощо. Але потребує підписки при масштабі роботи.

Для EfficientNet Lite, обидва інструменти підійдуть, але CVAT буде кращим вибором.

Процес розмітки.

Необхідні зображення 3D-друкованих об'єктів, як з дефектами, так і без.

Розмітку зробити використовуючи обраний інструмент (LabelImg або CVAT), розмітити області на зображеннях, які містять дефекти.

Важливо створити чіткі, точні анотації, оскільки якість розмітки безпосередньо впливає на ефективність навчання моделі.

Підготовка датасету для навчання.

Для EfficientNet Lite рекомендується використовувати формат TFRecord, оскільки він ефективно працює з TensorFlow і дозволяє зручно масштабувати процес навчання.

Після розмітки конвертувати дані та анотації в формат TFRecord.

Для цього можна використовувати скрипти конвертації, які часто доступні в репозиторіях, пов'язаних з TensorFlow.

Для конвертації анотацій з формату PASCAL VOC (XML-файли, які генерує LabelImg) у формат TFRecord, який підходить для тренінгу моделей в TensorFlow, можна використати наступний процес:

TensorFlow зазвичай надає скрипти для конвертації датасетів з різних форматів у TFRecord.

Можливо знайти такий скрипт або створити власний. Стандартний скрипт для конвертації може бути знайдений у репозиторіях TensorFlow, наприклад, у TensorFlow Model Garden.

Може знадобитися встановити додаткові бібліотеки для виконання скрипта, наприклад lxml та Pillow: “pip install lxml pillow”

Скрипт зазвичай вимагатиме вказування шляху до папки з XML-файлами, шляху до зображень та вихідного шляху для TFRecord.

Запустити скрипт, вказавши необхідні параметри.

Приклад команди для скрипта конвертації:

```
“python generate_tfrecord.py --csv_input=data/train_labels.csv --
image_dir=images/train --output_path=train.record”
```

Цей приклад вказує на використання скрипта generate_tfrecord.py, який конвертує дані з формату CSV у TFRecord. Потрібно буде адаптувати команду під конкретний скрипт та структуру директорій.

Залежно версії TensorFlow та конкретних потреб, скрипти та їх параметри можуть відрізнятися.

Після створення файлів TFRecord, треба переконайтись, що вони коректно читаються TensorFlow. Це можна зробити, протестувавши їх у тренувальному пайплайні.

Тренувальний, тестовий та валідаційний набори:

Розділити датасет на тренувальний, тестовий та валідаційний набори.

Зазвичай, розподіл відбувається у співвідношенні 70% (тренування), 20% (валідація), 10% (тестування).

3.6 Навчання моделі

Конфігурація тренування – перед початком тренування моделі потрібно правильно встановити гіперпараметри, які впливають на процес навчання та якість кінцевої моделі.

Гіперпараметри – це зовнішні конфігураційні налаштування, які встановлюють до початку процесу навчання моделі. Вони відрізняються від параметрів, які модель оптимізує під час тренування (наприклад, ваги у нейронній мережі). Гіперпараметри керують загальним процесом навчання моделі, і їх правильний вибір може мати значний вплив на ефективність навчання та якість моделі.

Серед гіперпараметрів є:

1. Швидкість навчання (Learning Rate) – це один з ключових параметрів, який впливає на те, наскільки швидко модель "вчиться". Занадто висока швидкість може призвести до нестабільного навчання, а занадто низька – до повільного навчання.

2. Розмір пакету (Batch Size) – визначає кількість зразків даних, що обробляються перед оновленням внутрішніх параметрів моделі. Великий розмір пакету може прискорити тренування, але збільшує вимоги до пам'яті.

3. Кількість епох (Epochs) – визначає, скільки разів модель буде тренуватися на всьому Dataset. Недостатність епох може призвести до недонавчання, а забагато – до перенавчання.

4. Розмір сітки (Grid Size) у згорткових нейронних мережах впливають на кількість параметрів у згорткових шарах.

5. Кількість прихованих шарів та нейронів у повністю з'єднаних шарах: для глибоких нейронних мереж можливо визначити, скільки шарів та нейронів використовувати.

6. Функція активації: визначає вихідні значення нейронів (наприклад, ReLU, Sigmoid, Tanh).

7. Втрати (Loss Function): функція, яка вимірює різницю між передбаченнями моделі та фактичними даними.

8. Оптимізатор (Optimizer): алгоритм, що використовується для оновлення ваг моделі під час тренування (наприклад, Adam, SGD).

Для налаштування процесу тренування використовується бібліотека TensorFlow:

```
“model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001,
    loss='binary_crossentropy',
    metrics=['accuracy'])”
```

Після налаштування гіперпараметрів, модель готова до тренування.

Тренування моделі:

Використовуючи метод `fit` для тренування моделі на підготовлених даних:

```
“history = model.fit(x_train, y_train,
    batch_size=1,
    epochs=10,
    validation_data=(x_val, y_val))”
```

`x_train` та `y_train` – тренувальні дані та мітки відповідно, `x_val` та `y_val` – валідаційні дані.

Валідація допомагає оцінити ефективність моделі на даних, які не були використані під час тренування.

Використання валідаційного набору:

```
“model.fit(x_train, y_train, epochs=10,
    validation_data=(x_val, y_val))”
```

Валідаційний набір допомагає ідентифікувати проблеми, такі як перенавчання.

Аналіз результатів валідації:

Після кожної епохи оцінюються втрати та точність на валідаційному наборі.

Якщо втрати на валідаційному наборі зростають, тоді можливе перенавчання моделі.

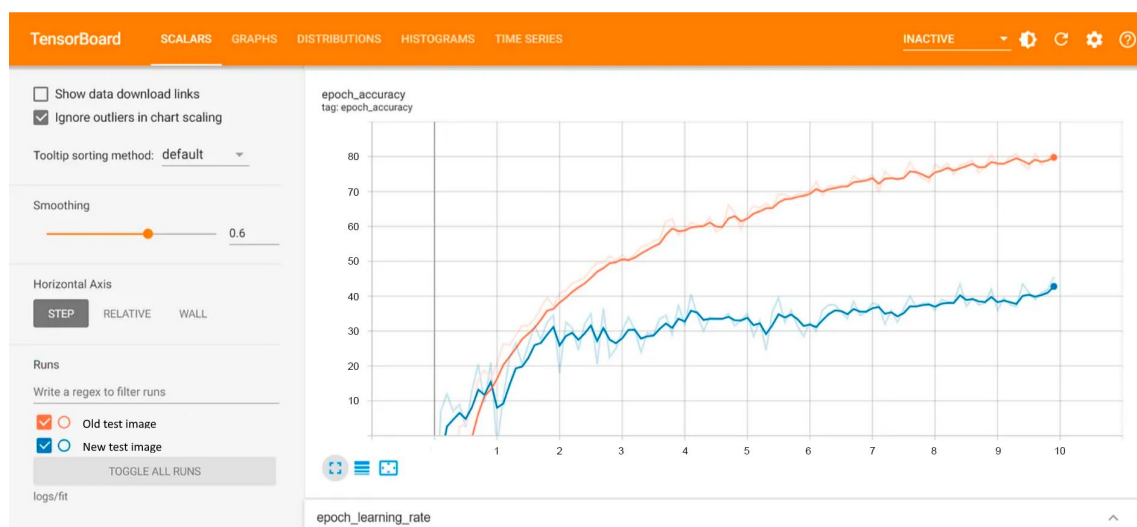
По завершенню тренування, аналізуються загальні показники ефективності моделі, використовуючи тестовий набір даних. Це допоможе оцінити, наскільки добре модель узагальнює навчання на нових даних.

Необхідно, використовувати TensorBoard або інші засоби для візуалізації прогресу навчання.

Навчання моделі – це ітераційний процес. Доведеться кілька разів повторити тренування з різними гіперпараметрами для досягнення найкращих результатів.

Тестові навчання моделі показують різні результати. Наприклад, якщо пройти 10 епох навчання на одній бібліотеці даних, тоді повторюється загальне правило, модель стає спеціалістом у даних дефектах і дедалі гірше аналізує нові раніше невідомі дефекти.

Результатом тестового навчання став графік у TensorBoard, який показує, що ефективність детектування відомих у навчанні дефектів росте, тоді як, ефективність знаходження нових дефектів перестає збільшуватись.



3.14. Таблиця ефективності навчання у процентах через епохи.

Після того як нейромережа з машинним зором була попередньо навчана на першому наборі даних, з'явилась можливість ідентифікувати дефекти друку.

В данному випадку було швидше та ефективніше навчити нейромережу ідентифікувати дефекти які є об'єктами, а саме - Spagetti. Вид дефекту друку який утворюється при друці без поверхні для адгезії, що було зумовлено результатом неправильного налаштування файлу інструкцій чи падіння підтримки.

Матриця помилок (confusion matrix) — це таблиця, яка використовується для оцінки ефективності моделі класифікації, зокрема в машинному навчанні та комп'ютерному зорі. Ця матриця дозволяє візуально представити, наскільки добре модель розрізняє класи.

Компоненти матриці помилок:

1. Істинно позитивні (TP - True Positives)

Визначення: випадки, коли модель правильно передбачає позитивний клас. Наприклад, якщо модель виявляє дефект на фотографії, і дефект є, це вважається істинно позитивним результатом.

Приклад: модель виявляє дефект друку на фотографії, і насправді дефект присутній.

2. Хибно позитивні (FP - False Positives)

Визначення: випадки, коли модель неправильно передбачає позитивний клас. Тобто модель вказує на наявність дефекту, коли насправді його немає.

Приклад: модель виявляє дефект на фотографії, але на ділі ця ділянка фотографії без дефекту.

3. Істинно негативні (TN – True Negatives)

Визначення: випадки, коли модель правильно передбачає негативний клас. Тобто модель вказує на відсутність дефекту, і він справді відсутній.

Приклад: модель не виявляє дефекту на ділянці фотографії, дефекту там дійсно немає.

4. Хибно негативні (FN - False Negatives)

Визначення: випадки, коли модель невірно передбачає негативний клас.

Тобто модель не виявляє дефект, хоча він насправді є.

Приклад: модель не виявляє дефекту на фотографії, але дефект там є.

Структура матриці:

Матриця помилок зазвичай має 2x2 формат для задач бінарної класифікації:

Значення матриці помилок:

Розуміння точності моделі: дозволяє оцінити, наскільки точно модель класифікує випадки.

Виявлення зміщень: помагає зрозуміти, чи є у моделі схильність до ложнопозитивних або ложнонегативних передбачень.

Поліпшення моделі: на основі цієї інформації можна вносити корективи в модель для підвищення її точності та надійності.

Підрахування:

Істинно позитивні (TP): кількість випадків, коли модель правильно передбачає позитивний клас.

Хибно позитивні (FP): кількість випадків, коли модель неправильно передбачає позитивний клас.

Істинно негативні (TN): кількість випадків, коли модель правильно передбачає негативний клас.

Хибно негативні (FN): кількість випадків, коли модель неправильно передбачає негативний клас.

Оцінка точності:

- **Підрахунок правильних виявлень:** для кожної фотографії підрахуйте скільки дефектів було правильно виявлено (справді позитивні результати, TP).
- **Підрахунок пропущених дефектів:** визначте, скільки дефектів не виявила модель (хибно негативні результати, FN).

- **Підрахунок помилкових виявлень:** вкажіть, скільки разів модель помилково помітила не дефектні ділянки як дефектні (хибнопозитивні результати, FP).

F1-бал (F1 Score) — це статистична метрика, яка використовується для оцінки точності бінарної класифікації моделей, особливо у випадках, коли класи не збалансовані. F1-бал є гармонічним середнім точності (precision) та повноти (recall), яке дозволяє об'єднати ці дві метрики в одну для більшої зручності порівняння та аналізу.

Формула F1-бала: $F1 = 2 \times \text{precision} \times \text{recall} / (\text{precision} + \text{recall})$

Розрахунок метрик:

Точність (Precision): Відношення правильно виявлених дефектів до загальної кількості виявлених дефектів.

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

Повнота (Recall): Відношення правильно виявлених дефектів до кількості дефектів на фотографіях.

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

Тестування комп'ютерного зору

Для тестового набору було відібрано 10 фотографій дефектів друку.

Загалом 20 об'єктів на фотографіях, нейромережа знайшла 16 об'єктів, з них 2 знайдені помилково, 6 дефектів пропущено.

Точність (Precision): 0.875

0.875 — це відсоток правильно знайдених дефектів з усіх знайдених.

Повнота (Recall): 0.7

0.7 — це відсоток правильно знайдених дефектів з усіх існуючих дефектів.

F1-бал: 0.778

0.778 — це гармонічне середнє точності та повноти, яке допомагає оцінити баланс між ними.

Таким чином перший набір для навчання через 10 епох дав результат детекції більший за 50 відсотків.

3.7 Дослідження параметрів процесу управління 3D принтером

Параметри управління 3D принтером з використанням машинного навчання та без нього представлено в таблиці 3.3.

Таблиця 3.3

Порівняння параметрів управління 3D принтером з використанням машинного навчання та без нього

Параметр	З машинним навчанням	Без машинного навчання
Якість друку	Покращена	Залежить від статичних налаштувань або загальних рекомендацій
Час друку	Швидший	Може бути менш ефективним без адаптивних параметрів
Витрати матеріалів	Зменшені	Можуть бути вищими без оптимізації параметрів друку
Складність управління	Може вимагати складніші алгоритми, але може бути більш автоматизованим	Може бути менш складним, але менше автоматизованим

Таблиця демонструє різницю у параметрах управління 3D принтером з використанням машинного навчання та без нього, вказуючи на переваги та недоліки кожного підходу за кількома ключовими аспектами.

Розглянемо метрики оцінки якості друку 3D моделей.

Метрики та їх формули для оцінки якості друку на основі методів обробки та аналізу зображень 3D моделей:

1. Peak Signal-to-Noise Ratio (PSNR):

$$PSNR = 10 \cdot \lg \left(\frac{x_{\max}^2}{\bar{x}} \right) \quad (3.3)$$

де x_{\max} – максимально допустиме значення, \bar{x} – середньоквадратичне значення помилки

Ця метрика вимірює відношення максимально можливого сигналу до рівня шуму, що присутній у сигналі. В контексті друку 3D вона може використовуватися для оцінки точності відтворення деталей моделі.

2. Structural Similarity Index (SSIM):

SSIM враховує 3 складові: яскравість (Luminance), контрастність (Contrast) та структурна подібність (Structure):

1. Яскравість (Luminance):

- Цей аспект відображає різницю у яскравості між пікселями.
- В індексі SSIM він представлений як $l(x,y)$, де x та y – це піксельні значення двох зображень. Ця складова оцінюється шляхом врахування середньоквадратичної відстані між цими значеннями.

2. Контрастність (Contrast):

- Ця частина оцінює різницю в контрастності між зображеннями.
- У формулі SSIM це представлено як $c(x,y)$, де x та y – це середнє значення контрасту для двох зображень.

3. Структурна подібність (Structure):

- Цей аспект відображає подібність структур об'єктів на зображеннях.
- У формулі SSIM він представлений як $s(x,y)$, де x та y – це коваріація між значеннями пікселів для двох зображень.

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (3.4)$$

де $l(x,y)$ - метрика яскравості; $c(x,y)$ - метрика контрастності; $s(x,y)$ - метрика структурної подібності; α, β, γ - вагові коефіцієнти для кожної метрики відповідно. Зазвичай вони можуть мати значення близькі до 1.

SSIM порівнює два зображення, оцінюючи схожість між ними. В контексті друку 3D це може використовуватися для визначення, наскільки точно відтворено деталі моделі.

3. Час друку:

$$T = \frac{k \cdot t_k}{v} \quad (3.5)$$

де k – кількість шарів, t_k – час друку одного шару, v – швидкість друку.

Ця формула розраховує загальний час друку для обраної моделі, враховуючи кількість шарів, час друку одного шару та швидкість друку.

4. Витрати матеріалу:

$$P = V \cdot \rho \quad (3.6)$$

де V – об'єм друку, ρ – густина матеріалу.

Ця формула розраховує кількість матеріалу, який буде використано для друку обраної моделі на основі об'єму друку та густини матеріалу.

Таблиця 3.4

Розрахунки метрик для процесу друку з використанням та без використання машинного навчання

Метрика	Без машинного навчання	З машинним навчанням
PSNR	52.61	51.93
SSIM	0.82	0.88
Час друку (хвилини)	52.5	36
Витрати матеріалу (г)	1200	1080

Ці метрики і формули використовуються для оцінки якості та ефективності процесу друку 3D. Вони допомагають визначити точність відтворення моделі, час та ресурси, які витрачаються під час друку.

В таблиці 3.4 представлено розрахунки метрик для процесу друку з використанням та без використання машинного навчання.

Ці дані дозволяють порівняти результати для кожної метрики між процесом друку з використанням машинного навчання та без нього.

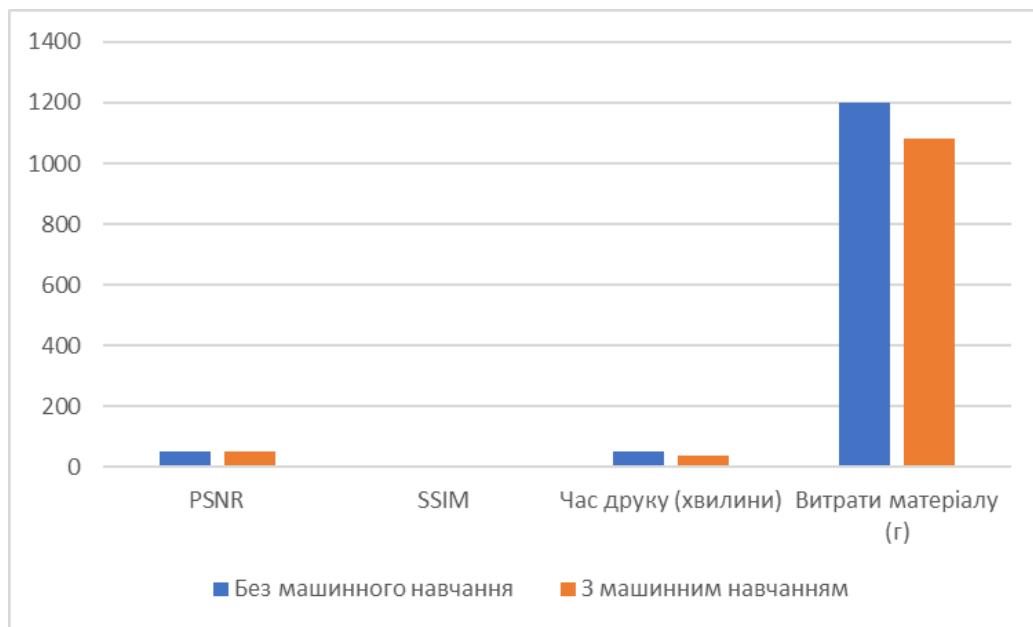


Рис. 3.15. Розрахунки метрик для процесу друку з використанням та без використання машинного навчання

Проаналізувавши отримані значення можемо прийти до наступного висновку:

1. PSNR (Peak Signal-to-Noise Ratio):

- **Без машинного навчання:** значення PSNR складає близько 52.61, що вказує на високу якість відтворення друку без використання машинного навчання.
- **З машинним навчанням:** значення PSNR близько 51.93, що може вказувати на меншу точність відтворення друку з використанням машинного навчання, але різниця не є суттєвою.

2. SSIM (Structural Similarity Index):

- **Без машинного навчання:** SSIM близько 0.82, що показує наявність значних різниць між оригінальним та відтвореним зображеннями.
- **З машинним навчанням:** SSIM наближається до 0.88, що вказує на кращу схожість між оригінальним та відтвореним зображеннями при використанні машинного навчання.

3. Час друку:

- **Без машинного навчання:** час друку становить приблизно 52.5 хвилини, що може бути вищим через меншу ефективність оптимізації параметрів.
- **З машинним навчанням:** час друку скоротився до приблизно 36 хвилин, що свідчить про більшу ефективність використання оптимальних параметрів завдяки машинному навчанню.

4. Витрати матеріалу:

- **Без машинного навчання:** витрати матеріалу становлять 1200 г, але можуть бути вищими через меншу ефективність оптимізації.
- **З машинним навчанням:** витрати матеріалу зменшилися до 1080 г, що свідчить про кращу оптимізацію використання матеріалу завдяки машинному навчанню.

Отже, машинне навчання в даному випадку дозволило поліпшити схожість зображення, скоротити час друку та зменшити витрати матеріалу, що може вказувати на загальну покращену ефективність процесу друку 3D моделей.

3.8 Перший результат навчання нейромережі

Для навчання нейромережі яка використовує комп'ютерний зір для пошуку дефектів 3D друку, була використана бібліотека TensorFlow. Також був використаний програмний засіб як середовище розробки під назвою "PyTorch" який мав власний інструментарій для розмітки дефектів на зображеннях "LabelImage".

Після створення датасету з матеріалами для навчання був наступний етап, який полягав у запуску нейромережі та початку навчання, яке використовувало потужності центрального процесору для роботи середовища, а для навчання нейромережі були використані потужності дискретної відеокарти, з причини її більшої потужності, та відмінного від процесору способу обробки даних та алгоритмів роботи архітектури, через що вона ефективніше оброблювала дані і досягала результату швидше.

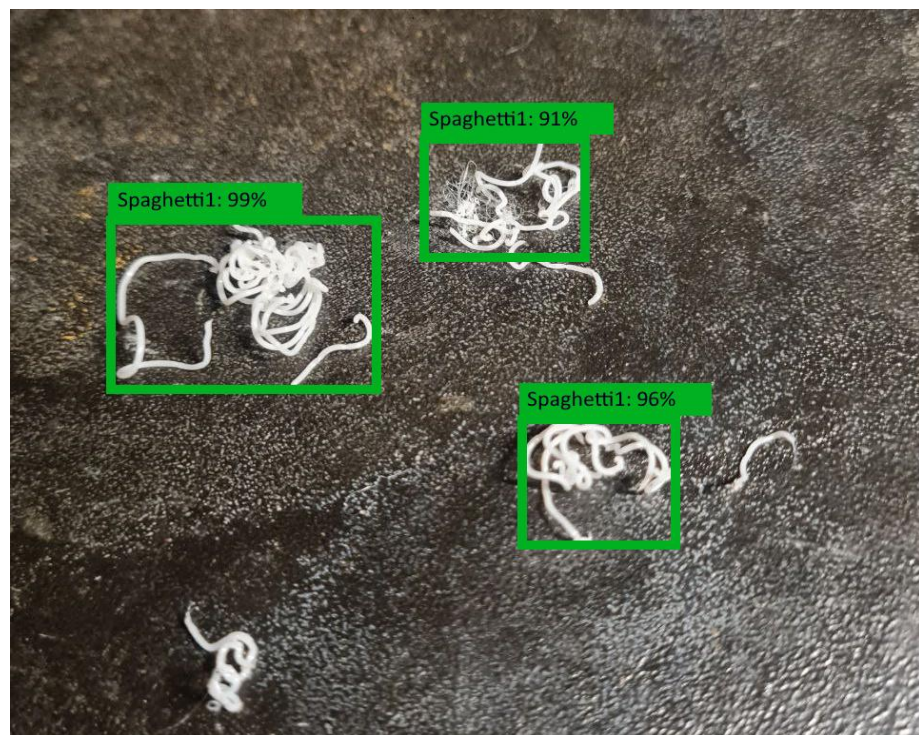


Рис. 3.16. Процес роботи нейромережі під час пошуку дефектів друку.

На зображенні видно, що нейромережа, запущена на комп'ютері використовуючи камеру, змогла ідентифікувати дефекти друку з певною точністю і впевненістю.

Нейромережа розпізнала перший дефект друку, який називають спагеті, розпізнала не всі через примітивність навчання та якість зображення. Також присутні проценти які повідомляють про впевненість мережі, що саме цей об'єкт є дефектом.

Нейромережа здатна ідентифікувати два типи дефектів, перший дефект – павутиння, він же волосіння та спагеті. Дефект який часто утворюється в результаті

неправильного налаштування параметрів друку. Наприклад, коли в слайсері не додали підтримки на нависаннях і окремих елементах, які розділені один від одного та не мають основи на якій починається їх пошаровий друк.

Також, це стосується і внутрішніх елементів, впадини також можуть друкуватись без основи, якщо не використовується заповнення моделі. В результаті чого утворюються елементи нестабільного вигляду, через що деякі елементи моделі не друкуються, і матеріал витрачений на них буде втрачено без толку, також можливий прямий вплив на всю модель, після чого друк буде зіпсовано, а час та матеріал втрачено в пусту.

Нейромережа навчена знаходити подібні структури з великою точністю, завдяки їх сильній різниці у формах. Також, якщо використовувати режим великої ефективності обробки, у вигляді обробки зображення у реальному часі, де нейромережа аналізує від одного кадру на секунду, тоді на одних кадрах дефект може бути не знайдено, але на великій кількості кадрів дефект буде розпізнано, в такому випадку дефект буде ідентифіковано статистично.

Другий тип дефектів – недостатня екструзія. В даному випадку навчання було складним через меншу помітність дефекту, необхідність у кращій якості зображення, та необхідність нових етапів навчання з новими даними.

Модель необхідно продовжувати навчати, покращуючи здатність знаходити вже відомі типи дефектів, і навчати знаходити нові типи дефектів друку.

Насправді, якщо розробляти робота як повноцінний проект, тоді для виявлення візуально помітних дефектів друку необхідно навчити модель знаходити більше 10 типів помилок друку, модель в такому випадку збільшить свою вагу, та почне обробляти зображення під більшу частину фільтрів, використовуючи все більше ресурсів процесору для своєї роботи. В такому випадку з'явиться необхідність використовувати більш потужний комп'ютер, наприклад, наступну версію Распаберрі, чи перейти на портативний комп'ютер на 64 розрядній системі, в такому випадку програма вже буде працювати не на ARM архітектурі, і знадобиться окрема переробка програмних засобів.

ВИСНОВКИ

1. В процесі науково дослідницької роботи розроблений прототип нейромережі, яка здатна ідентифікувати дефекти друку використовуючи комп'ютерний зір.
2. Створена перша версія головного веб-інтерфейсу для контролю параметрів роботи принтеру та для контролю через камеру.
3. Додатково розроблений прототип мобільного додатку, який копіює функціонал веб-інтерфейсу.
4. Ведеться розробка власного проекту роботу-маніпулятора на основі відкритого джерела, досліджений спосіб підключення та керування.
5. Узагальнені способи навчання нейромережі для подальшої розробки та впровадження фізичний прототип системи.
6. Проведена робота по налаштуванню операційної системи, яка надає доступ до функціоналу 3D принтеру.
7. Прописаний алгоритм дій для програми при ідентифікуванні дефектів друку, з використанням функцій які сприймає контроллер принтеру через функціонал OctoPrint.
8. Написаний тестовий код на Python, який реагує на ідентифікацію дефекту і згідно з алгоритмом дій.

ПЕРЕЛІК ПОСИЛАНЬ

1. Керівництво з FDM 3D-друку [Електронний ресурс] – Режим доступу до ресурсу: <https://3d4u.com.ua/uk/blog/post/88-d-druk-dlya-pochatkivciv-pochatok-roboti-z-3d-drukum>.
2. Про Raspberry Pi [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Raspberry_Pi.
3. Про програмне забезпечення для Raspberry Pi [Електронний ресурс] – Режим доступу до ресурсу: <https://raspberrypi.org/software>.
4. Використання OctoPrint з Raspberry Pi для керування 3D-принтерами [Електронний ресурс] – Режим доступу до ресурсу: <https://www.raspberrypi.com/news/octoprint/>.
5. Офіційні дані OctoPrint [Електронний ресурс] – Режим доступу до ресурсу: <https://octoprint.org/>.
6. Посібник щодо створення веб-інтерфейсу для керування Raspberry Pi [Електронний ресурс] – Режим доступу до ресурсу: <https://www.aranacorp.com/en/create-a-web-interface-to-control-your-raspberry-pi/>.
7. Зашифроване керування віддаленими серверами [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bitvise.com/ssh-client>.
8. Інструкції щодо створення домашньої системи безперебійного живлення [Електронний ресурс] – Режим доступу до ресурсу: <https://www.homemade-circuits.com/making-ups-tutorial/#:~:text=URL%3A%20https%3A%2F%2Fwww.homemade>.
9. Модуль контролю між роботом та платою комп'ютера [Електронний ресурс] – Режим доступу до ресурсу: <https://www.nxp.com/products/power-management/lighting-driver-and-controller-ics/led-controllers/16-channel-12-bit-pwm-fm-plus-ic-bus-led-controller:PCA9685>.
10. Інструкція створення керування роботом [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/stawo/robot-arm>.
11. Альтернативний проект маніпулятора [Електронний ресурс] – Режим доступу до ресурсу: <https://hackaday.io/project/25895-6dof-robotic-arm>.
12. Проект роботу-маніпулятора [Електронний ресурс] – Режим доступу до ресурсу: <https://www.thingiverse.com/thing:1229690>.
13. Інструкції та рекомендації щодо використання G-кодів [Електронний ресурс] – Режим доступу до ресурсу: <https://3dpworld.co/>.
14. Докладний список команд G-коду [Електронний ресурс] – Режим доступу до ресурсу: <https://all3dp.com/2/3d-printer-g-code-commands-list-tutorial/>.

15. Помилки 3D-друку [Електронний ресурс] – Режим доступу до ресурсу: <https://berserk.design/uk/pomilki-3d-druku-chomu-voni-trapliaiutsia-i-iak-chasto>
16. Стаття, в якій обговорюється впровадження комп'ютерного зору та штучного інтелекту AI Build для контролю якості у 3D-друку [Електронний ресурс] – Режим доступу до ресурсу: <https://3dprintingindustry.com/news/ai-build-introduces-computer-vision-for-3d-printing-quality-control-156704/#:~:text=Umair%20Iftikhar%20June%2010th%202019,vision%20and%20AI%20C%20the.>
17. Стаття про використання штучного інтелекту (AI) для виявлення помилок у 3D-друку в реальному часі [Електронний ресурс] – Режим доступу до ресурсу: <https://www.3dnatives.com/en/ai-build-3d-printing-errors-110620195/#:~:text=An%20obstacle%20to%20large,and%20warping%20in%20real%20time.>
18. Machine Learning with PyTorch and Scikit-Learn / Sebastian Raschka., 2022. – 770 с.
19. Програмування на Python [Електронний ресурс] – Режим доступу до ресурсу: <https://diveinto.org/python3/>.
20. EfficientNet-Lite [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.tensorflow.org/2020/03/higher-accuracy-on-vision-models-with-efficientnet-lite.html>.
21. Курс по програмуванню нейромережі. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.avt.global/cv>.
22. Перелік книг та курсів для вивчення робототехніки [Електронний ресурс] – Режим доступу до ресурсу: <https://www.imena.ua/blog/most-popular-sources-for-robotics/>.
23. Глибоке навчання [Електронний ресурс] – Режим доступу до ресурсу: <https://www.coursera.org/specializations/deep-learning>.
24. Стаття EfficientNet [Електронний ресурс] – Режим доступу до ресурсу: <https://arxiv.org/abs/1905.11946>.
25. Репозиторій TensorFlow Lite на GitHub [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/models>.
26. Практичне навчання з використанням TensorFlow [Електронний ресурс] – Режим доступу до ресурсу: <https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/>.
27. Офіційна документація TensorFlow [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tensorflow.org/guide>.
28. Репозиторій III [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/TheSpaghettiDetective/obico-server>.

29. Документація для алгоритмів ШІ [Електронний ресурс] – Режим доступу до ресурсу: <https://pytorch.org/>.
30. Бібліотека OpenAI Gym для навчання і тестування алгоритмів ШІ [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/openai/gym>.
31. Тьюторіал Tensorflow [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tensorflow.org/tutorials/images>.
32. Комп'ютерний зір [Електронний ресурс] – Режим доступу до ресурсу: <https://www.udacity.com/course/neural-networks-for-computer-vision--ud810>.
33. PrintRite3D від Sigma: інструмент якості [Електронний ресурс] – Режим доступу до ресурсу: <https://www.3dnatives.com/en/ai-softwares-in-3d-printing2007234/>.
34. ШІ для пошуку дефектів [Електронний ресурс] – Режим доступу до ресурсу: <https://3dprint.com/265545/spaghetti-detective-open-source-ai-software-detects-prints-gone-wrong/>.
35. Використання глибокого навчання для детекції дефектів [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mdpi.com/2571-5577/4/2/34>.
36. Глибоке навчання для пошуку дефектів [Електронний ресурс] – Режим доступу до ресурсу: <https://www.emerald.com/insight/content/doi/10.1108/RPJ-05-2023-0157/full/html>.
37. Стаття навчання Комп'ютерного зору. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mdpi.com/2227-9717/8/11/1464>.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

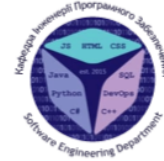
(Презентація)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Магістерська робота

«ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОЦЕСУ УПРАВЛІННЯ 3D ПРИНТЕРОМ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ»

Виконав: студент групи ПДМ-63 Олейніков Іван Анатолійович

Керівник: д.т.н., проф., завідувач кафедри ТЦР Жебка Вікторія Вікторівна

Київ - 2024

МЕТА, ОБ'ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: Розробка системи контролю друку з використанням машинного зору.

Об'єкт дослідження: Ідентифікація дефектів друку. Алгоритми дій. Способи мережевого керування. Спеціалізоване програмне забезпечення.

Предмет дослідження: моделі штучного інтелекту, необхідне апаратне забезпечення, алгоритми ідентифікації дефектів, алгоритми дії при дефектах друку. Веб-інтерфейс.

Порівняння доступних моделей машинного зору

Назва	Особливості	Переваги	Недоліки
MobileNet	спеціалізується на мінімізації обчислювального навантаження. Вона використовує глибокі роздільні згортки для зменшення кількості параметрів та операцій.	є дуже ефективною для мобільних пристроїв та інших ситуацій з обмеженими обчислювальними ресурсами.	може відставати від більш потужних моделей, таких як EfficientNet, з точки зору точності у складних завданнях комп'ютерного зору.
ShuffleNet	використовує унікальну техніку змішування каналів для підвищення ефективності згорткових операцій.	є надзвичайно легкою та ефективною з точки зору обчислювальних витрат, що робить її ідеальною для пристроїв з обмеженими ресурсами.	Подібно до MobileNet, може поступатися у точності більш складним моделям.
SqueezeNet	відома своєю екстремальною ефективністю у використанні параметрів, використовуючи "fire modules" - спеціальні блоки, які зменшують кількість параметрів.	ає дуже маленький розмір, що робить її підходящою для вбудованих систем та мобільних додатків.	маючи дуже малий розмір, вона може не досягати такої ж точності, як більш великі та складні моделі.
EfficientNet Lite	спрощена версія оригінальної моделі, призначена для використання на пристроях з обмеженими обчислювальними ресурсами.	менші системні вимоги порівняно з повною версією, більша швидкодія, легкість інтегрування	Менш точний від оригіналу, менші можливості пошуку, менше можливостей для масштабування.

3

Дефекти та дії після ідентифікування

Дії при дефектах		
№	Назва	Вирішення
1	Недостатня екструзія (або відсутня)	Нестача матеріалу. Збільшити подачу - Код: M221 S[відсоток] (більше ніж 100)
2	Надмірна екструзія	Надлишок подачі філаменту. Зменшити подачу - Код: M221 S[відсоток] (менше 100)
3	Павутиння	Марна подача матеріалу у повітря. Ігнорувати. Призупити - Код: "M0"/"M25"
4	Викривлення	Збільшити обдув - Код: M106 S255 ; Сповільнення друку - Код: M220 S50
5	Зміщення слоїв	Проблема механіки. Призупинити/Закінчити друк - Код: M0/M112
6	Вібрації слоїв	Велика швидкість. Сповільнити друк - Код: M220 S50
7	Перегрів	Зменшення температури - Код: M104 S[температура];
8	Помилка першого слою	Сопло близько до столу. Підняти - Код: G1 Z[значення в мм]; Сопло високо над столом. Опустити - Код: G1 Z-[значення в мм]

4

Алгоритм навчання:



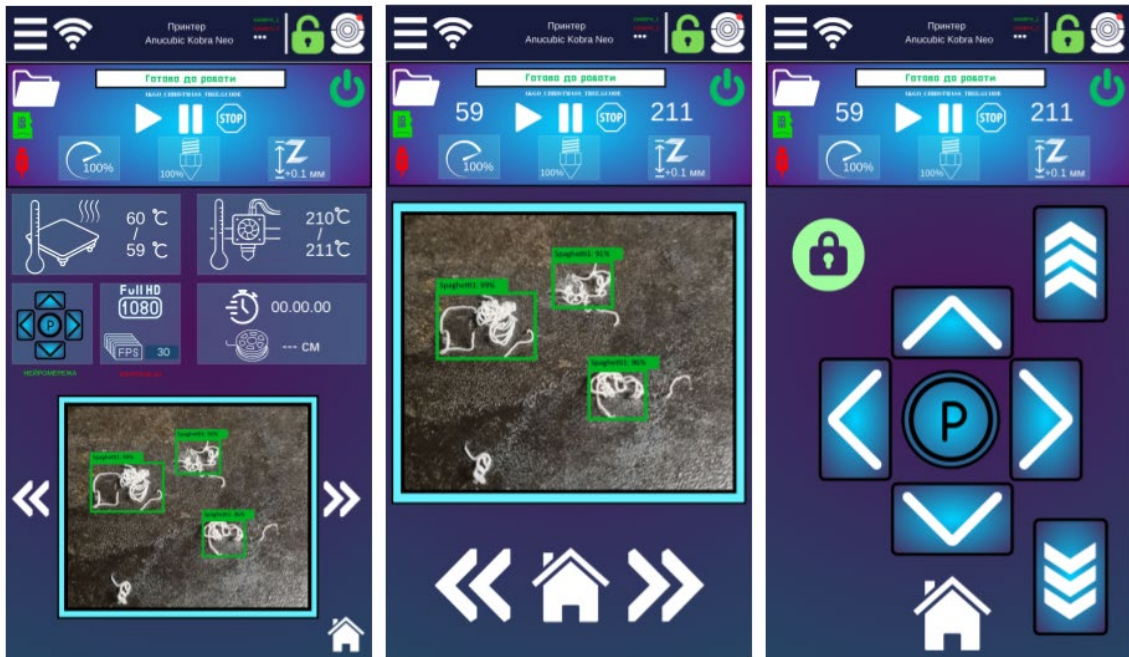
5

Прототип веб-інтерфейсу.



6

Інтерфейс мобільного застосунку



7

Перевірка прогресу навчання.



8

Результат навчання комп'ютерного зору



9

ВИСНОВКИ

1. В процесі науково дослідницької роботи розроблений прототип неймережі яка здатна ідентифікувати дефекти друку використовуючи комп'ютерний зір.
2. Створена перша версія головного веб-інтерфейсу для контролю параметрів роботи принтеру та для контролю через камеру.
3. Додатково розроблений прототип мобільного додатку, який копіює функціонал веб-інтерфейсу.
4. Ведеться розробка власного проекту роботу-маніпулятора на основі відкритого джерела, досліджений спосіб підключення та керування.
5. Узагальнені способи навчання неймережі для подальшої розробки та впровадження фізичний прототип системи.
6. Проведена робота по налаштуванню операційної системи, яка надає доступ до функціоналу 3D принтеру.
7. Прописаний алгоритм дій для програми при ідентифікуванні дефектів друку, з використанням функцій які сприймає контроллер принтеру через функціонал OctoPrint.
8. Написаний тестовий код на Python, який реагує на ідентифікацію дефекту і згідно з алгоритмом дій.

10

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Статті:

1. Олейніков І.А. Підвищення ефективності процесу управління 3D принтером з використанням додаткового обладнання та методів машинного навчання // Телекомунікаційні та інформаційні технології № 1(83), 2024 (прийнята до друку)

Тези доповідей:

1. Олейніков І.А. Підвищення ефективності процесу управління 3D принтером з використанням машинного навчання. // Науково-практична конференція «проблеми комп'ютерної інженерії». – Київ: ДУТ, 2023. – (Тези прийняті, збірник ще не опублікований)

2. Олейніков І.А. Підвищення ефективності процесу управління 3D принтером з використанням машинного навчання. // І Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу». – Київ: ДУТ, 2023. – (Тези прийняті, збірник ще не опублікований)

11

ДЯКУЮ ЗА УВАГУ!

12