

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ
Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту
Завідувач кафедри

_____ Олена ОЛЬХОВСЬКА
(підпис)

«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему

**«Програмна реалізації відтворення зображення
для верстату по виготовленню трикотажних виробів»**

зі спеціальності 122 «Комп'ютерні науки»
освітня програма «Комп'ютерні науки»
ступеня бакалавра

Виконавець роботи

Матвієнко Володимир Володимирович

_____ «___» _____ 2025 р.

(підпис)

Науковий керівник

к.ф.-м.н., доц., Чілікіна Тетяна Василівна

_____ «___» _____ 202_ р.

(підпис)

Рецензент

ПОЛТАВА 2025

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ Олена ОЛЬХОВСЬКА

«_____» вересня 202__ р.

ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему «Програмна реалізації відтворення зображення для верстату по виготовленню трикотажних виробів»

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня бакалавра

Прізвище, ім'я, по батькові Матвієнко Володимир Володимирович

Затверджена наказом ректора № ____ -Н від «__» _____ 202__ р.

Термін подання студентом роботи «__» _____ 2025_ р.

Вихідні дані до кваліфікаційної роботи: технічне завдання від підприємства.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Графічний матеріал

ВСТУП

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Ознайомлення з процесом обробки фото на підприємстві

2.2 Огляд вимог до нейромережі для розв'язання поставлених завдань

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Аналіз вимог до програми.

3.2 Створення дашборду в середовищі Google Таблиці

3.3 Алгоритм роботи програми

3.4 Вдосконалення та генерація зображень за допомогою нейронної мережі

3.5 (4.5) Алгоритм роботи нейронної мережі

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1 Визначення а створення структурних складових програми

4.2 Створення класів з методами

4.3 Блок-схема роботи програми

4.4 Додавання методів для взаємодії з нейромережею

4.5 Тестування функціональності готового продукту

ВИСНОВКИ

СПИСОК ЛІТЕРАТУРИ

ДОДАТОК А

Консультанти розділів кваліфікаційної роботи

Розділ	ПІБ, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постановка задачі	Чілікіна Т.В.		
Інформаційний огляд	Чілікіна Т.В.		
Теоретична частина	Чілікіна Т.В.		
Практична реалізація	Чілікіна Т.В.		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику		
3. Постановка завдання		
4. Інформаційний огляд джерел бібліотек та інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		
8. Доповідь студента на кафедрі		
9. Доопрацювання (за необхідності), рецензування		

Дата видачі завдання « ____ » _____ 202__ р.

Здобувач вищої освіти Матвієнко Володимир Володимирович

Науковий керівник к. ф.-м. н., Чілікіна Тетяна Василівна

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____

(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № ____ від « ____ » _____ 202__ р.

Секретар ЕК _____

(підпис)

(ініціали та прізвище)

Затверджую
 Керівник _____
 к.ф.-м.н. Тетяна Чілікіна
 «_____» _____ 2025 р

План

Дипломного проєкту з фаху

спеціальності 122 Комп'ютерні науки

освітня програма 122 Комп'ютерні науки

на тему «Програмна реалізації відтворення зображення для верстату по виготовленню трикотажних виробів»

Прізвище, ім'я, по батькові Матвієнко Володимир Володимирович
 ВСТУП

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Ознайомлення з процесом обробки фото на підприємстві

2.2 Огляд вимог до нейромережі для розв'язання поставлених завдань

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Аналіз вимог до програми.

3.2 Створення дашборду в середовищі Google Таблиці

3.3 Алгоритм роботи програми

3.4 Вдосконалення та генерація зображень за допомогою нейронної мережі

3.5 Алгоритм роботи нейронної мережі

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1 Визначення а створення структурних складових програми

4.2 Створення класів з методами

4.3 Блок-схема роботи програми

4.4 Додавання методів для взаємодії з нейромережею

4.5 Тестування функціональності готового продукту

ВИСНОВКИ

СПИСОК ЛІТЕРАТУРИ

ДОДАТКИ

Здобувач вищої освіти Матвієнко Володимир
 «_____» _____ 202_ р

(підпис)

РЕФЕРАТ

Записка: 50 с., 5 рис., 1 таблиця, 11 джерел.

КЛАСТЕРИЗАЦІЯ, СПРОЩЕННЯ ЗОБРАЖЕНЬ, ТРИКОТАЖНИЙ ВЕРСТАТ, АЛГОРИТМИ ОБРОБКИ.

Об'єкт розробки — програмна реалізація відтворення зображення для верстату, який використовується у виробництві трикотажних виробів.

Мета роботи — розробка алгоритмів обробки та спрощення зображень для їх подальшого використання на верстаті по виготовленню трикотажних виробів. Алгоритми мають забезпечувати зниження складності зображень, що дозволить верстату коректно обробляти їх і використовувати для виробництва.

Методи дослідження — методи кластеризації та спрощення зображень (k-means, спрощення кольорової палітри, заміна кольорів), технології квантування кольору та побудови графічних алгоритмів обробки зображень для текстильних виробів.

У роботі здійснено огляд існуючих підходів до обробки зображень для текстильного обладнання, розглянуто можливості автоматизованих верстатів, проаналізовано ефективні алгоритми кластеризації та спрощення для зменшення кольорової глибини зображень. Було розроблено покроковий алгоритм обробки зображення, який включає спрощення кольорової палітри, кластеризацію кольорів, а також адаптивну обробку для оптимального друку на трикотажних виробах.

В результаті програмної реалізації створено прототип інтерфейсу, який дозволяє обирати та обробляти зображення для друку на верстаті з різними налаштуваннями спрощення і контролю якості. Прототип було протестовано для забезпечення якості друку при різних параметрах кольорової палітри і деталізації. Результати розробки можуть бути використані у виробничих процесах текстильної промисловості для автоматизації підготовки зображень на трикотажних верстатах.

Ключові слова: трикотажний верстат, генерація зображень, кластеризація, нейронна мережа, Stable Diffusion, спрощення зображень, адаптація, обробка графіки, візуалізація, автоматизація, Java, Google Sheets, квантування кольорів.

ЗМІСТ

ВСТУП

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Ознайомлення з процесом обробки фото на підприємстві

2.2 Огляд вимог до нейромережі для розв'язання поставлених завдань

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Аналіз вимог до програми.

3.2 Створення дашборду в середовищі Google Таблиці

3.3 Алгоритм роботи програми

3.4 Вдосконалення та генерація зображень за допомогою нейронної мережі

3.5 (4.5) Алгоритм роботи нейронної мережі

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1 Визначення а створення структурних складових програми

4.2 Створення класів з методами

4.3 Блок-схема роботи програми

4.4 Додавання методів для взаємодії з нейромережею

4.5 Тестування функціональності готового продукту

ВИСНОВКИ

СПИСОК ЛІТЕРАТУРИ

ДОДАТКИ

ВСТУП

Актуальність. У сучасному в'язаному виробництві автоматизація процесів відіграє важливу роль, дозволяючи досягти високої точності та ефективності. Використання інформаційних технологій і спеціалізованих програм для керування обладнанням значно скорочує час виготовлення виробів і значно знижує витрати, не жертвуючи якістю. Впровадження програмного забезпечення для виробничих машин стало нагальною потребою — воно сприяє підвищенню якості продукції, звільненню кадрових ресурсів та зміцнює конкурентні позиції текстильних підприємств.

Метою роботи є створення елементів програмного забезпечення для відображення та управління виготовленим в'язаним вихором на машині. Програма повинна значно спростити процес роботи над підбором нитководів та кольорів. А також спростити роботу дизайнерів та художників.

Об'єктом розробки в даній роботі є програмний інтерфейс для верстата по виготовленню трикотажних виробів.

Предметом розробки є алгоритми та візуальні елементи для відображення процесу виробництва трикотажних виробів. А також його симуляції, для практичного не унеможливлення браку.

Головне завдання – розробити систему, яка дозволить візуалізувати хід виробництва трикотажних виробів, надаючи користувачу інтуїтивно зрозумілий інтерфейс для взаємодії з верстатом.

Перелік використаних методів полягає у застосуванні методів об'єктно-орієнтованого програмування, комп'ютерної графіки та алгоритмів обробки даних для забезпечення точного відображення та контролю виробничих процесів.

При реалізації системи використано мову програмування Python та бібліотеки для роботи з графічними інтерфейсами, що дозволяє створити зручний для користувача візуальний компонент для моніторингу виробництва.

Робота складається з чотирьох розділів. У першому розділі розглядаються основні етапи автоматизації виробничих процесів та програмні засоби, які використовуються для керування текстильними верстатами. У другому розділі описані особливості розробки програмного забезпечення для візуалізації виробничого процесу. У третьому розділі відображений процес підготовки до роботи та пошук важливої інформації. У четвертому розділі детально описана розробка програми та ключові розроблені блоки коду.

Обсяг пояснювальної записки: 54 стор., в т.ч. основна частина – 30 стор., джерела – 15 назв.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

З метою автоматизації процесу створення дизайнів шкарпеток і підвищення ефективності підготовки виробничих даних було поставлено задачу розробити програмний продукт, який дозволить конвертувати графічні зображення у формат, придатний для подальшого використання шкарпетковим верстатом. Основною функцією програми є обробка будь-якого вхідного зображення, його спрощення до форми, яку здатна обробити машина, та подальше перетворення у 16-бітне BMP-зображення. Цей BMP-файл потім може бути конвертований у машинний код за допомогою сторонніх програм.

Основна мета розробки полягає у створенні зручної у використанні програми, яка зможе приймати довільні зображення та перетворювати їх у спрощену форму з урахуванням технічних обмежень верстата. Програма повинна автоматично аналізувати зображення, обирати прийнятну палітру кольорів, спрощувати графіку таким чином, щоб вона залишалась впізнаваною, але водночас могла бути відтворена верстатом із обмеженими можливостями. Також передбачено створення схеми, яка візуально відображає нитководи — елементи машини, що керують кольоровими нитками. Кожна горизонтальна лінія зображення повинна бути супроводжена вказівкою, які нитководи активуються, і ці елементи відображаються різними кольорами для зручності візуального аналізу.

Для значного спрощення роботи дизайнера та підвищення якості автоматизованої обробки зображень у програму інтегровано неймережу Stable Diffusion. Завдяки цьому користувач отримує можливість генерувати ескізи або стилізовані зображення за текстовим описом або на основі вже існуючих зображень. Це дозволяє дизайнерам не лише пришвидшити процес створення нових візуальних концепцій, а й зменшити потребу в складному ручному редагуванні. Неймережа автоматично адаптує візуальний стиль, згладжує переходи, спрощує форми й враховує обмеження, накладені машинною інтерпретацією. Така інтеграція значно

знижує бар'єр входу для користувачів без художньої освіти й надає широкі можливості для експериментів з дизайном.

Під час реалізації програми враховано важливі технічні обмеження. Верстат підтримує лише 16 нитководів загалом, але на кожній горизонтальній лінії зображення дозволяється використовувати не більше шести одночасно. До того ж один із кольорів повинен залишатися незмінним на всьому зображенні — як правило, він виконує роль основного або фонового кольору. Це обмеження вимагає реалізації спеціального алгоритму оптимізації палітри, який здатен визначати ключові кольори зображення таким чином, щоб максимально зберегти візуальні деталі, не надмірно спрощуючи графіку.

Особливу увагу приділено простоті використання програми. Вона має бути зрозумілою навіть для недосвідчених користувачів персонального комп'ютера. Інтерфейс програми повинен бути інтуїтивно зрозумілим, із підказками та поясненнями до кожного елементу керування. Це дозволить швидко опанувати програму і розпочати роботу без додаткового навчання або залучення фахівців.

Крім того, програма повинна підтримувати стандартні формати зображень, такі як JPEG, PNG, BMP, і містити вбудовані інструменти для базового редагування: обрізання, масштабування, зміну яскравості та контрастності. Це забезпечить користувачам можливість підготувати зображення без використання сторонніх графічних редакторів, роблячи процес максимально автономним і зручним.

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Ознайомлення з процесом обробки фото на підприємстві

Цей інформативний огляд має на меті не лише надати докладний аналіз ринку програмних продуктів, що допомагають оптимізувати та спростити виробничі процеси, але й надати поглиблений аналіз основних тенденцій, технологічних інновацій та сучасних підходів, що стимулюють розвиток цієї галузі. Огляд охоплює широкий спектр рішень, від програмного забезпечення для автоматизації виробничих ліній до систем управління підприємством, що інтегрують новітні технології, такі як штучний інтелект, машинне навчання та Інтернет речей (IoT). Важливим аспектом є визначення ефективності цих інструментів у реальних виробничих умовах, їхнього впливу на зниження витрат, покращення якості продукції та скорочення часу, потрібного для виконання операцій.

Огляд, представлений у цьому дослідженні, стане корисним інструментом для підприємств, які прагнуть зрозуміти, які програмні рішення найкраще відповідають їхнім реальним потребам та стратегічним цілям. Він допоможе не лише обрати оптимальні інструменти, а й оцінити сильні та слабкі сторони вже наявних систем. Це, своєю чергою, відкриє шлях до вдосконалення процесів, інтеграції новітніх технологій і підвищення загальної ефективності виробництва.

Сфера трикотажного виробництва має вагоме значення для текстильної промисловості як в Україні, так і за її межами. Вона охоплює різноманітну продукцію — від повсякденного одягу до товарів для дому. Проте саме виробництво шкарпеток вирізняється стабільним зростанням і постійним попитом. Шкарпетки, як базовий елемент гардероба, швидко зношуються та часто потребують оновлення, що ставить перед виробниками завдання постійно адаптуватися до змін ринку. Це спонукає до впровадження інновацій — як у виборі матеріалів, так і в організації виробничих процесів.

В Україні виробництво трикотажу, зокрема шкарпеток, має всі передумови для активного розвитку. Однак українським виробникам доводиться долати чимало труднощів: сильна конкуренція з імпортом, обмеженість інвестицій у новітнє обладнання та потреба в автоматизації виробничих процесів, щоб підвищити ефективність. Попри це, дедалі більше споживачів віддають перевагу вітчизняним товарам, що свідчить про зростання інтересу до української продукції на тлі загальної підтримки національного виробника.

За наявними даними, український ринок шкарпеткової продукції демонструє стабільне зростання на рівні близько 10% щорічно. Виробники активно вкладають кошти в оновлення та модернізацію виробничих ліній, прагнучи забезпечити високі стандарти якості та зміцнити свої позиції як на внутрішньому, так і на зовнішньому ринках. Серед провідних тенденцій галузі варто виокремити орієнтацію на екологічність: усе більшого значення набуває використання натуральних і перероблених матеріалів, а також впровадження інноваційних технологій, що дають змогу виготовляти зручні, міцні й повітропроникні вироби.

В умовах жорсткої конкуренції цифровізація стає одним із вирішальних чинників розвитку української шкарпеткової індустрії. Завдяки впровадженню сучасних цифрових технологій підприємства мають змогу не лише оптимізувати витрати, а й помітно підвищити продуктивність та якість готової продукції. Розглянемо основні етапи виробництва шкарпеток, де застосування цифрових рішень може дати найбільший ефект.

1. Контроль якості на кожному етапі виробництва

Щоб забезпечити стабільно високу якість продукції, сучасні підприємства впроваджують автоматизовані системи контролю. Вони значно знижують імовірність появи браку, забезпечують єдиний стандарт якості на всіх етапах виробництва та дають змогу швидше виявляти й усувати недоліки. Крім того, автоматизація зменшує вплив людського чинника, що позитивно позначається

на загальній ефективності процесу. У підсумку це не лише економить ресурси на виправлення помилок, а й підвищує довіру споживачів до бренду.

2. Автоматизація управління обладнанням

Використання спеціалізованого програмного забезпечення для керування в'язальним обладнанням значно спрощує контроль за його роботою. Системи дозволяють в реальному часі виявляти збої, оперативно їх усувати та точно налаштовувати параметри відповідно до вимог конкретної моделі шкарпеток. Такий рівень автоматизації не лише оптимізує роботу машин, а й забезпечує стабільну якість продукції, прискорюючи виробничий цикл і зменшуючи ризики простоїв.

3. Аналіз даних та оптимізація процесів

Збирання та аналіз даних на кожному етапі виробничого процесу дають можливість об'єктивно оцінювати ефективність використання ресурсів, рівень браку, а також витрати часу й матеріалів. Використання аналітичного програмного забезпечення допомагає виявляти «вузькі місця» в технологічному ланцюгу, що дозволяє приймати обґрунтовані управлінські рішення для подальшої оптимізації. Сучасний розвиток виробничих підприємств тісно пов'язаний із впровадженням новітніх інформаційних технологій, які сприяють зниженню витрат, вдосконаленню процесів і стабільному забезпеченню високої якості продукції. У галузі трикотажного виробництва, зокрема у виготовленні шкарпеток, автоматизація відіграє ключову роль, дозволяючи українським компаніям залишатися конкурентоспроможними як на внутрішньому, так і на світовому ринках.

Уявіть собі фабрику шкарпеток майбутнього, де замість нескінченного контролю вручну кожна нитка й кожен стібок під'єднані до єдиної «мозкової мережі» виробництва. Ця мережа постійно збирає дані з усіх вузлів — від натягу ниток у машині до вологості приміщення й енерговитрат. Завдяки аналітиці в реальному часі така система здатна:

- **Самостійно виявляти й запобігати дефектам**, перш ніж вони переростуть у масштабну проблему;

- **Оптимізувати налаштування обладнання**, зменшуючи споживання електроенергії та знижуючи витрати на технічне обслуговування;
- **Прогнозувати попит** і підлаштовувати виробничі лінії під конкретні партії замовлень.

Для українських виробників це не просто «модний тренд» — це шанс вийти на новий рівень конкурентоспроможності. Сьогодні, коли імпортні шкарпетки штурмують полиці супермаркетів низькими цінами, наше завдання — відрізнятись вищою якістю, унікальним дизайном і відповідальністю перед довкіллям. ІТ-рішення допомагають:

1. **Швидко впроваджувати інновації:** інтеграція з існуючими лініями не вимагає довгих простоїв — за лічені тижні ви отримуєте «розумний конвеєр», готовий до нових експериментів із матеріалами й формами.
2. **Показувати прозорі звіти:** клієнти та партнери бачать вуглецевий слід, енергоспоживання та рівень утилізації відходів — і вибирають тих, хто дбає про екологію.
3. **Запускати нові серії зі швидкістю стартапу:** від ідеї до першої пробної партії може проходити час, порівнянний із розробкою мобільного додатку.

В результаті, замість консервативного заводу понад сто років тому, українська шкарпеткова галузь отримує «**розумний цех**», де кожен процес і кожна деталь — від нитки до упаковки — працюють злагоджено і під контролем штучного інтелекту. Це не просто про економію: це про творчий підхід до традицій, про швидке реагування на запити ринку й про ту саму людяність, яка виражається в турботі про споживача та планету. Завдяки цифровим рішенням наші бренди зможуть створювати шкарпетки, які будуть не лише зручними, а й стануть символом відповідального виробництва та інноваційних ідей.

Уявіть собі фабрику шкарпеток майбутнього, де замість нескінченного контролю вручну кожна нитка й кожен стібок під'єднані до єдиної мозкової мережі виробництва. Ця мережа постійно збирає дані з усіх вузлів — від

натягу ниток у машині до вологості приміщення й енерговитрат. Завдяки аналітиці в реальному часі така система здатна:

- самостійно виявляти й запобігати дефектам, перш ніж вони переростуть у масштабну проблему
- оптимізувати налаштування обладнання, зменшуючи споживання електроенергії та знижуючи витрати на технічне обслуговування
- прогнозувати попит і підлаштовувати виробничі лінії під конкретні партії замовлень

Для українських виробників це не просто модний тренд — це шанс вийти на новий рівень конкурентоспроможності. Сьогодні, коли імпорتنі шкарпетки штурмують полиці супермаркетів низькими цінами, наше завдання — відрізнитися вищою якістю, унікальним дизайном і відповідальністю перед довкіллям. IT-рішення допомагають:

1. швидко впроваджувати інновації: інтеграція з існуючими лініями не вимагає довгих простоїв — за лічені тижні ви отримуєте розумний конвеєр, готовий до нових експериментів із матеріалами й формами
2. показувати прозорі звіти: клієнти та партнери бачать вуглецевий слід, енергоспоживання та рівень утилізації відходів — і вибирають тих, хто дбає про екологію
3. запускати нові серії зі швидкістю стартапу: від ідеї до першої пробної партії може проходити час, порівняний із розробкою мобільного додатку

В результаті, замість консервативного заводу понад сто років тому, українська шкарпеткова галузь отримує розумний цех, де кожен процес і кожна деталь — від нитки до упаковки — працюють злагоджено і під контролем штучного інтелекту. Це не просто про економію: це про творчий підхід до традицій, про швидке реагування на запити ринку й про ту саму людяність, яка виражається в турботі про споживача та планету. Завдяки цифровим рішенням наші бренди зможуть створювати шкарпетки, які будуть не лише зручними, а й стануть символом відповідального виробництва та інноваційних ідей.

2.2 Огляд вимог до нейромережі для розв’язання поставлених завдань

У процесі розробки та вдосконалення застосунку виникла необхідність автоматизувати роботу із зображеннями. Багато користувачів витрачають значний час на пошук або створення графіки вручну — процес, який часто є довгим і незручним.

Було вирішено зробити весь процес максимально простим і приємним для користувача, тому переглянули кожен його крок із точки зору автоматизації та зручності. Наша ідея — не просто додати можливість швидкого створення зображень, а організувати справжнє “творче середовище”, де користувач миттєво перетворює свої думки на візуальні образи. Для цього ми інтегрували нейронну мережу, здатну за лічені секунди згенерувати картинку за текстовим запитом, і водночас створили комфортний інтерфейс, який практично не потребує інструкцій.

Уявіть собі: ви вводите опис бажаного зображення — від загальних настроєвих фраз на кшталт “ранковий туман у лісі” до детальних вказівок про колір, текстуру та стиль — і натискаєте одну кнопку. Нейромережа працює “за лаштунками”, замінюючи десятки рутинних кроків (пошук відповідного референсу, налаштування фільтрів, підбір шрифтів тощо). Результат з’являється миттєво в тому самому вікні, де ви можете одразу внести правки: змінити яскравість, адаптувати композицію, підкоригувати відтінки або додати дрібні елементи.

Ключове завдання нашої команди полягало в тому, щоб весь інструментарій працював як єдине ціле. Немає потреби паралельно відкривати кілька програм чи вкладок браузера — жодних зайвих переходів і завантажень. Все, що необхідно для створення, редагування й експорту графічного контенту, розташовано в одному вікні: панель налаштувань генератора, попередній перегляд, інструменти доопрацювання деталей і кнопка збереження в потрібному форматі.

Такий підхід кардинально змінює робочий потік. Замість ручного експериментування з окремими фільтрами ви можете пройти шлях від ідеї до готового зображення за кілька кліків. Це заощаджує час не лише творчим професіоналам, які працюють над рекламними макетами або ілюстраціями, а й студентам і новачкам, яким важливо швидко побачити результат своїх задумів та отримати натхнення для подальших експериментів.

У перспективі ця платформа легко масштабується: ми вже продумали можливість додавання нових моделей нейромереж, підтримки різних стилів (від акварелі до 3D-рендерів) і інтеграції з іншими сервісами (наприклад, стоковими бібліотеками чи засобами колаборації в команді). Усе це робить наш застосунок не просто інструментом для генерації картинок, а справжньою творчою лабораторією, де будь-яка ідея швидко оживає на екрані.

На початковому кроці ми розглянули кілька варіантів реалізації цієї технології. Один із них полягав у створенні власної нейронної мережі з нуля, спеціально для наших потреб. Теоретично це могло б дати нам максимальну гнучкість і точність при виконанні завдань, але на практиці виявилось, що цей підхід потребує надзвичайно великих ресурсів. Тренування нейронної мережі вимагає величезних обсягів даних, потужного апаратного забезпечення та багато часу, а також залучення експертів з машинного навчання. Крім того, підтримка та подальше оновлення такої нейронної мережі стало б окремим складним завданням, що могло б значно сповільнити розробку програми в цілому.

Було розглянута можливість використання сторонніх нейронних мереж, які вже доступні через відкриті API. З одного боку, цей варіант здався привабливим, оскільки він зменшує навантаження на пристрій користувача, оскільки всі обчислення виконуються на сервері постачальника послуг. Однак з іншого боку, такі рішення мають свої обмеження. Багато з них не мають достатньо широкого функціоналу, що може завадити нам реалізувати всі необхідні функції. Крім того, умови їхнього комерційного використання не

завжди є зрозумілими, а залежність від зовнішніх сервісів створює додаткові ризики, пов'язані з їхньою стабільністю.

Зрештою, після детального аналізу всіх можливих варіантів і порівняння різних нейронних мереж, ми дійшли висновку, що найкращим рішенням для нашого застосунку є використання Stable Diffusion. Ця нейронна мережа довела свою ефективність як потужний інструмент обробки зображень, здатний генерувати графіку високої якості при мінімальних витратах ресурсів. Однією з основних переваг Stable Diffusion є її відкритий вихідний код, що дозволяє нам гнучко налаштовувати і адаптувати її під специфічні потреби нашого застосунку, а також інтегрувати її з іншими компонентами системи без суттєвих обмежень. Крім того, Stable Diffusion має широкий спектр функціональних можливостей, включаючи здатність генерувати зображення на основі текстових запитів, стилізацію зображень і можливість працювати безпосередньо з графічними даними в реальному часі. Це забезпечує велику гнучкість у роботі з різними типами зображень, що є важливим для нашого застосунку. Активна підтримка цієї нейронної мережі великою спільнотою розробників гарантує постійний розвиток і оновлення технології, що дозволяє нам підтримувати високу якість і відповідність сучасним вимогам. Отже, вибір Stable Diffusion був оптимальним, оскільки ця технологія не лише відповідає всім нашим вимогам, але й відкриває нові можливості для подальшого розвитку та вдосконалення застосунку.

Stable Diffusion має низку особливостей, які роблять її ідеальним вибором для нашого проєкту. По-перше, це відкритий вихідний код, що означає, що ми можемо вільно адаптувати цю технологію до наших потреб без обмежень і додаткових витрат. По-друге, нейронна мережа активно розвивається, і велика кількість ентузіастів постійно працює над її вдосконаленням, створенням нових моделей, оптимізацією швидкості роботи та додаванням нових корисних функцій. Це гарантує, що ми завжди матимемо доступ до останніх оновлень і зможемо використовувати найсучасніші технології.

Ще однією важливою перевагою Stable Diffusion є її гнучкість. Користувач може самостійно налаштовувати параметри генерації зображень, вибирати стиль, рівень деталізації та експериментувати з різними моделями, що дозволяє отримувати максимально точні та якісні результати. Також ця неймережа має широкий вибір способів взаємодії через API, що дозволяє нам інтегрувати її в нашу програму без зайвих труднощів, спрощуючи процес обробки зображень і роблячи його більш зручним та ефективним.

Завдяки інтеграції Stable Diffusion в нашому програмному забезпеченні ми отримуємо низку суттєвих переваг. По-перше, значно прискорюється робота з графікою, оскільки користувачам більше не потрібно витратити час на пошук або редагування зображень вручну. По-друге, це підвищує зручність використання – тепер всі необхідні інструменти доступні в одному застосунку, що усуває необхідність перемикатися між програмами та спрощує робочий процес. По-третє, це розширює можливості для користувачів – вони можуть експериментувати з різними параметрами та отримувати результати, що найкраще відповідають їхнім потребам.

Крім того, Stable Diffusion є відкритою технологією, що дає значну перевагу порівняно з іншими комерційними рішеннями, оскільки її можна використовувати без обмежень у комерційних проєктах. Це означає, що ми маємо повну свободу в реалізації та розширенні функціональності нашого програмного забезпечення без необхідності враховувати обмеження, пов'язані з ліцензійними умовами сторонніх послуг або продуктів. Така відкритість дозволяє нам вільно адаптувати алгоритми, налаштовувати їх під специфічні вимоги проєкту та впроваджувати нові функції без побоювань щодо порушення авторських прав або необхідності отримання додаткових дозволів. Це також дозволяє уникнути фінансових витрат на ліцензування комерційних продуктів і дає можливість оптимізувати витрати на розробку. Оскільки технологія активно підтримується великою спільнотою розробників, ми маємо доступ до постійно оновлюваних інструментів, що дозволяє нам швидко впроваджувати останні досягнення в галузі машинного навчання та

комп'ютерного зору. Як результат, ми можемо не лише значно покращити якість нашого продукту, але й залишатися гнучкими у відповіді на зміни на ринку та потреби користувачів. Відкритість Stable Diffusion також забезпечує стабільність та перспективи довгострокового розвитку нашого програмного забезпечення, оскільки ми не залежимо від зовнішніх постачальників і можемо самостійно керувати всіма аспектами його роботи.

У нашому додатку ми зробили сміливий вибір на користь Stable Diffusion — і не просто так. Ця модель дозволяє оживляти ідеї миттєво: натиснув кнопку — і готове зображення в твоїх руках, без марудних налаштувань і постійних “переходів” між різними програмами. Відкритий вихідний код дає свободу: хочеш адаптувати алгоритм під власні задачі — будь ласка, і жодних витрат на ліцензії чи прихованих платежів.

У наступних оновленнях ми перетворимо інструментарій у справжню лабораторію візуальних експериментів. Заплановано відразу три категорії моделей: базові “усе-в-одному” для швидкого результату, творчі режими з арт-фокусом і “дивацтві” — бета-версії з несподіваними налаштуваннями, які пробудять художника в кожному. Плюс з'являться інтегровані коректори кольору, алгоритми підвищення різкості й автоматичного очищення шумів — щоб не відриватися від творчого потоку.

Ми не зупиняємося й на цьому. Оптимізація продуктивності — наш наступний трюк: підтримка GPU-прискорення та розподілених обчислень дозволить обробляти десятки запитів одночасно, скорочуючи час генерації до мінімуму. А поєднання з хмарними платформами та спільними робочими просторами зробить колаборацію миттєвою — ділитися результатами можна буде за кілька кліків.

Оскільки Stable Diffusion щомиті оновлюється завдяки відкритій спільноті, наш додаток завжди матиме доступ до найсвіжіших алгоритмів і фішок. Ми гарантуємо: з кожним релізом він ставатиме розумнішим, потужнішим і ще ближчим до потреб користувачів, залишаючись при цьому надзвичайно простим і зручним.

РОЗДІЛ 3. Теоретична частина

3.1 Аналіз вимог до програми

Після вивчення та спостереження за виробничими процесами було з'ясовано, що для підготовки зображень до друку дизайнер вручну обробляє фотографії та стежить за тим, щоб вони відповідали технічним обмеженням швейної машини. Найважливішим обмеженням для машини є те, що на одну лінію друку шкарпеток можна використовувати лише п'ять кольорів, а загальна кількість кольорів для одного зображення не повинна перевищувати чотирнадцяти.

Обмеження катушок з нитками суттєво впливають на фінальний вигляд дизайну, оскільки вони визначають доступну кольорову палітру та здатність точно передавати деталі зображення. Загалом є шість напрямних для ниток, і кожна має свої особливості.

C1 і C2 можуть обробляти по два кольори кожна, що робить їх корисними для базових елементів дизайну, таких як фони або контури. Проте вимкнення цих елементів суттєво зменшує доступну палітру, що може ускладнити роботу зі складнішими зображеннями. У таких випадках необхідно адаптувати зображення, спростивши його або змінивши контрастність, щоб воно залишалось чітким і зрозумілим в нових умовах. Це може включати зменшення кількості деталей або використання простіших форм для збереження ефективності та ясності.

Катушки з нитками C3, C4 і C5 підтримують до трьох кольорів, що надає більше гнучкості для створення багатоколірних дизайнів. Проте, якщо обмежити їх використання, дизайн може втратити важливі переходи та візуальні ефекти. У таких випадках потрібно буде адаптувати зображення,

перерозподілити кольори між доступними нитками або використовувати альтернативні методи, такі як точкові візерунки, щоб імітувати плавні градієнти.

F1 працює лише з одним кольором, тож має обмежене застосування, але чудово підходить для суцільних елементів, таких як фони, контури або окремі частини, які не потребують складних колірних переходів.

У контексті автоматизованої обробки зображень і підготовки до в'язання необхідно використовувати алгоритми, які ефективно адаптують зображення до заданих обмежень. Це можуть бути методи кластеризації кольорів, корекція палітри за допомогою нейронних мереж і модифікація зображення зі збереженням ключових візуальних елементів. Коли нейронна мережа інтегрована в програмне забезпечення для спрощення зображень, такі алгоритми можуть автоматично розподіляти кольори відповідно до можливостей ниток, що значно спрощує процес підготовки дизайну для в'язальної машини.

Уявіть собі ситуацію в'язальної майстерні, де нитки живуть власним життям: кожен їх відтінок не просто фарба, а особистість, зі своїми властивостями та уподобаннями. Саме ці “характерні риси” ниток диктують правила гри для нашого дизайнерського ПЗ. Якщо з'ясовується, що C1 чи C2 (або обидві) раптово “пішли на перерву”, програма мусить миттєво реагувати: перебудовувати палітру, перерозподіляти кольорові смуги між доступними котушками й навіть трансформувати орнаменти, щоб нова гама виглядала гармонійно та продуманою.

Цей процес — ніби жонглювання вогнями на сцені: картинка повинна залишатися чіткою, а лінії — плавними, хоча в запасі вже не всі яскраві барви. У такі моменти на перший план виходять гнучкі алгоритми, здатні “перекроювати” дизайн в реальному часі. Вони підраховують, які відтінки можна замінити, які фрагменти малюнка зберегти незмінними, а де варто ввести стилістичну “вправку”, щоб нейтралізувати відсутність кольору. Все це

відбувається автоматично, наче художник із швидкою рукою та гнучким оком, який миттєво підлаштовується під нові умови.

Насправді головна мета не лише зберегти технічну точність — важливо також зберегти душу зображення. Часом доводиться хитрувати: використовувати близькі відтінки для передачі глибини, або підсилювати контраст у деталях, щоб втрачені кольори не кидалися в очі. Результат — стабільний і плавний виробничий потік, де кожна пара шкарпеток виходить не просто добротною, а справжнім витвором мистецтва. Адже справжня інновація — це здатність адаптуватися, не втрачаючи натхнення і краси початкової ідеї.

Важливим компонентом є забезпечення високої якості відтворення, що включає точність у деталях і кольорах, навіть під час адаптації зображення до нових умов. Алгоритми повинні справлятися з різними рівнями складності й мінливості, щоб забезпечити найкращий можливий результат незалежно від змін у доступних ресурсах, дозволяючи досягати високої ефективності та відповідати потребам користувача без втрати якості.

Таким чином, завдання дизайнера полягає в попередній обробці фотографій, обмежуючи палітру до максимум 15 кольорів і оптимізуючи кожен рядок зображення так, щоб у ньому використовувалося не більше п'яти кольорів. Це вимагає постійного контролю та ручних коригувань, що може призводити до помилок або зниження продуктивності. У фінальній програмі важливо враховувати специфічні вимоги до обробки зображень для певних типів шкарпеток, наприклад, тих, що не мають окремих подавачів ниток. Якщо виникає необхідність вимкнути C1 або C1 і C2, це суттєво впливає на палітру доступних кольорів, оскільки зменшується кількість ниток, які можна використати для створення візерунків або градієнтів. Відповідно, це підвищує складність алгоритмів генерації зображення, адже вони мають адаптувати дизайн до обмеженої кількості кольорів.

Для вирішення цієї проблеми необхідно додати до фінальної програми додаткову логіку, яка забезпечить коректну роботу навіть за умов обмежень

ниток. Це включає алгоритми автоматичного підбору палітри та оптимізації візерунків, які зможуть ефективно використовувати обмежені ресурси з мінімальною втратою якості зображення на шкарпетках.

Отже, аналіз показав, що обмеження технічних можливостей друкарського обладнання та специфіка роботи з подавачами ниток є важливими факторами, які потрібно враховувати під час підготовки зображень для друку на шкарпетках. Кожен із подавачів ниток має певну пропускну здатність за кольором, що безпосередньо впливає на вибір палітри та складність дизайну. Вимкнення окремих подавачів (наприклад, C1 або C1 і C2) суттєво зменшує кількість доступних кольорів, що ускладнює автоматичне відтворення якісних візерунків.

Для спрощення процесу адаптації зображень до цих обмежень рекомендується створити застосунок, який зможе автоматично оптимізувати дизайн відповідно до доступної кольорової палітри та технічних обмежень для кожного рядка друку. Використання Java з Maven дасть змогу створити гнучку систему для автоматизації цього процесу, зменшуючи потребу в ручному втручанні та знижуючи ризик помилок.

Такий підхід підвищить продуктивність, забезпечить точність і значно спростить адаптацію зображень для друку навіть в умовах суворих обмежень за кольорами.

3.2 Створення дашборду в середовищі Google Таблиці

Для контролю процесу та виконання завдань було вирішено створити імпровізовану панель управління в Google Sheets. Такий вибір зумовлений гнучкістю Google Sheets, його доступністю для командної роботи в реальному часі та безкоштовністю, що робить цей інструмент зручним для організації та моніторингу всіх етапів проєкту без додаткових витрат. Приклад таблиці зображено на Рисунку 3.2.1

Виконавці	виконувач	треба зробити			Пояснення
Техно			Алгоритми:		
Фокс			Збільшення кількості кольорів	"++++"	
Равар	3	(Додав ефект стійка, сира версія)	Зменшити спрощення Перевірка алгоритму	"++++"	
			Зберігання поменню палітра	"++++"	
				"++++"	Для кольорів, що є у програми верстату
				"++"	
				"++"	
				"++"	Після редагування дизайнером
				"++"	Коректне форматування на 16 біт
				"++"	"+" заливка
			Класи:		
			-Фото(A)	2	Оператор для кольорів rgb, перетворення у 3 матриці
			-Конвертор(B)	1, 3, 4, 10	
			-Емулятор станка(C)	7, 9	Обмеження кількості допустимих нитководів!
			-Редагування фото(D)	5, 8, 12	
			Інтерфейс:		
			регулювання по ширині	A	Початкове, замовника, програми
				C	
				D	
				B	
				D	
				D	
			Юсер експірієнс		
			Встановлення		
			зміна назви	"++++"	

Рисунку 3.2.1 ДашБорд що використовувався

Крім того, панель дає змогу контролювати роботу двох розробників, забезпечуючи прозорість у розподілі та виконанні завдань, а також швидкий доступ до результатів кожного етапу обробки зображень. Це значно спрощує координацію роботи та дає можливість оперативно вносити зміни відповідно до потреб проєкту.

Створена панель охоплює всі основні етапи підготовки зображень до друку на шкарпетках — від початкового імпорту файлів до фінального коригування зображень перед передачею на друкарське обладнання. Вона слугує інтерфейсом, що суттєво спрощує робочий процес для дизайнерів і операторів друку, забезпечуючи зручне управління всіма аспектами обробки зображень.

Панель управління автоматизує ключові завдання, такі як зміна розміру, корекція кольору, налаштування контрасту та підготовка зображень до друку, що зменшує час на ручну обробку та знижує ймовірність помилок. Користувачі можуть швидко та ефективно адаптувати зображення відповідно до виробничих вимог, таких як обмеження палітри кольорів або специфікації для різних типів матеріалів, що використовуються у виробництві шкарпеток.

Крім того, панель забезпечує інтеграцію з іншими виробничими системами, що гарантує безперервність процесу — від розробки дизайну до готового продукту. Вона також дозволяє дизайнерам і операторам легко

відстежувати хід виконання завдань, вносити необхідні коригування та контролювати якість на кожному етапі, що підвищує ефективність роботи та забезпечує високу точність результату.

3.3 Алгоритм роботи програми

Робочий процес починається з конвертації файлів. Дизайнер завантажує зображення у будь-якому зручному форматі, а система автоматично перетворює його у формат, сумісний із друкарським обладнанням. Це дозволяє уникнути проблем із кольорами та роздільною здатністю. Наприклад, багато друкарських машин працюють із 16-бітними BMP-файлами, і ручна підготовка таких файлів займає багато часу. Панель бере на себе це завдання, автоматично налаштовуючи параметри зображення.

Після цього зображення проходить етап матричного подання. Це означає, що воно аналізується як набір пікселів у вигляді цифрової матриці, що полегшує подальший підбір кольорів. Оскільки друкарські машини мають обмеження на кількість кольорів у друку, дуже важливо правильно згрупувати відтінки, щоб зберегти якомога більше деталей, не перевищуючи технічні обмеження. Панель управління дозволяє автоматично оптимізувати палітру, вибираючи кольори, які найкраще підходять для друку.

Щоб уникнути невідповідностей у фінальному результаті, система також включає функцію емуляції друкарської машини. Вона дає змогу побачити, як виглядатиме зображення після друку ще до початку виробництва. Це дозволяє своєчасно виявити потенційні проблеми, такі як накладання кольорів, втрата деталей або невідповідність ниток.

Після проходження всіх перевірок зображення може бути масштабоване та відредаговане відповідно до точних розмірів продукту. Оператори можуть регулювати контраст, змінювати розміри та виконувати інші тонкі налаштування без необхідності використання сторонніх графічних редакторів.

На фінальному етапі панель управління генерує різні версії зображення для різних учасників процесу. Дизайнер отримує збережену версію для подальших змін, оператор друку — оптимізований файл у потрібному форматі, а клієнт — готовий макет для погодження.

Панелі управління значно скорочують час підготовки зображень, мінімізують помилки та автоматизують рутинні процеси. Це не тільки збільшує швидкість роботи, а й забезпечує стабільно високу якість друку.

Отже, панель управління в Google Sheets забезпечує повний цикл контролю — від конвертації файлів до передачі готових зображень. Використання Google Sheets також дозволяє швидко вносити коригування та співпрацювати над проєктом, що допомагає максимально підвищити продуктивність та командну роботу.

Для спільної розробки та контролю змін під час проєкту також використовувався GitHub. Цей інструмент дозволив ефективно відстежувати зміни в коді, організувати командну роботу та контролювати версії файлів. Як згадувалося раніше, було прийнято рішення розробити Java-додаток за допомогою Maven. Для цього ми обрали середовище розробки IntelliJ IDEA. Цей IDE надав зручність у написанні, відлагодженні та тестуванні коду завдяки потужним інструментам для роботи з проєктами Maven. Основні переваги IntelliJ IDEA в цьому проєкті:

1. **Інтеграція з Maven** — повна підтримка Maven дозволяє автоматично керувати залежностями, конфігураціями та збіркою проєкту, що значно спрощує налаштування середовища та забезпечує швидкий старт розробки.

2. **Зручність у написанні коду** — IntelliJ IDEA пропонує інтелектуальне автодоповнення, швидкі рефакторинги, попередження про можливі помилки та інші інструменти для підвищення продуктивності розробників.

3. **Вбудовані засоби налагодження** — IDE надає потужні інструменти для відстеження виконання коду в реальному часі, що дозволяє

швидко виявляти й виправляти помилки, особливо важливі для складних алгоритмів обробки зображень та оптимізації палітри.

4. **Інтеграція з GitHub** — завдяки інтеграції з GitHub, розробники можуть безпосередньо керувати репозиторіями з середовища IntelliJ IDEA, відстежувати зміни, створювати коміти та зливати гілки, що забезпечує зручний і безперервний робочий процес.

Отже, використання IntelliJ IDEA у поєднанні з Maven створило ефективне та гнучке середовище розробки, яке забезпечило всі необхідні інструменти для продуктивної роботи над проектом і спростило контроль якості коду та управління версіями.

У ході виконання цього проекту було створено імпровізовану панель управління в Google Sheets для моніторингу та контролю процесу підготовки зображень для друку на шкарпетках. Google Sheets став ефективним інструментом завдяки своїй гнучкості, безкоштовності та можливості роботи в реальному часі, що забезпечило зручний контроль за виконанням завдань, а також координацію між двома розробниками. Панель охоплює всі основні етапи обробки зображень: від конвертації файлів і підбору кольорів до емуляції друку та підготовки фінальних версій для різних користувачів.

Для спільної розробки коду та контролю змін використовувався GitHub, що дозволило ефективно відстежувати зміни, управляти версіями та координувати командну роботу. GitHub допоміг організувати надійний процес розробки, мінімізувати ризик конфліктів і забезпечити прозорість на кожному етапі.

Програмування на Java із використанням Maven стало важливою частиною цього проекту завдяки його перевагам в управлінні залежностями та автоматизації збірки. Maven забезпечує зручний спосіб визначення та керування бібліотеками, що використовуються в проекті, і автоматично завантажує їх із центрального репозиторію. Це мінімізує ймовірність помилок під час інтеграції сторонніх бібліотек і дозволяє командам швидко змінювати або оновлювати версії залежностей без додаткових зусиль.

Уявіть команду розробників, яка працює, як живий організм. Кожен шматок коду — мов клітина, що має свою функцію, а GitHub — це мозок, що координує їхню роботу. Тут немає плутанини з десятками версій файлів і невідомо ким зробленими правками. Кожна зміна має своє місце, свого автора і зрозумілу причину. Зробив коміт — і вже видно, що саме змінилось, навіщо це зроблено і як воно вплине на проєкт.

А коли команда працює над новою функцією, у гру вступає pull request. Це не просто технічний крок — це жива дискусія. Місце, де кожен може висловити свою думку, запропонувати покращення або просто звернути увагу на деталі. Один розробник помітить, що алгоритм можна зробити швидшим, інший — що назва змінної не зовсім зрозуміла, ще хтось згадає про тестування чи безпеку. І в підсумку ми отримуємо рішення, що пройшло через сито командного досвіду й критичного мислення.

Усе це супроводжується автоматизованими перевітками. Літери ловлять помилки в стилі коду, тести перевіряють, чи все працює як треба, а системи аналізу одразу попереджають про потенційні проблеми. GitHub навіть покаже, якщо кілька людей одночасно змінюють один і той самий файл — і допоможе розрулити конфлікти. Це не просто платформа — це ніби техлідер, який завжди поруч і готовий допомогти.

Коли всі зміни погоджено, і настав момент натиснути «Merge», це вже не просто технічна дія — це як фінальний штрих у спільній картині. Те, над чим ви працювали, стає частиною цілого. Ще один пазл ліг на своє місце. Команда рухається далі: нові ідеї, нові виклики, новий простір для розвитку.

GitHub у цьому процесі — не просто інструмент для збереження коду. Це місце, де народжується командна культура. Тут усе прозоро, чесно й відкрито. Можна сміливо експериментувати, бо є відчуття опори — будь-яку помилку можна виправити, а будь-яке рішення — обговорити. І саме завдяки цьому хаос сирих ідей поступово перетворюється на надійний і якісний продукт, за який не соромно.

Інтеграція з GitHub додає гнучкості та порядку: видно, хто що зробив, коли, навіщо й чому. Можна подивитися історію, порівняти версії, знайти джерело багу або навпаки — відкотитися до стабільного варіанту. Pull request'и — це можливість пройтися по коду з боку, кинути свіжий погляд, знайти потенційні слабкі місця ще до того, як зміни потраплять у головну гілку.

Цей процес дає простір для діалогу: залишити коментар, запропонувати покращення, поставити запитання. І все це — до релізу, не після. Коли кілька людей одночасно змінюють код, GitHub не просто повідомляє про конфлікти — він допомагає їх розв'язати. І ти не один з проблемою: є інструменти, є команда, є підтримка.

Ще одна сильна сторона — автоматичні перевірки. Кожен коміт одразу проходить тести, перевіряється стиль, знаходяться потенційні помилки. Це дозволяє не накопичувати проблеми, а вирішувати їх по ходу. І чим менше багів, тим спокійніший реліз, тим стабільніший продукт.

GitHub створює відчуття, що ти працюєш не просто з кодом, а в живому, організованому просторі, де кожен учасник — важлива частина команди. І де навіть найменша правка може стати початком чогось великого.

Усі ці інструменти та функції значно підвищили продуктивність команди та дозволили розробникам швидко адаптуватися до вимог проєкту. Вони могли зосередитися на досягненні кінцевої мети, а не витратити час на вирішення технічних проблем, які часто виникають під час налаштування середовищ або керування залежностями. Тому правильний вибір інструментів програмування не лише спростив процес розробки, а й забезпечив ефективну взаємодію команди та високу продуктивність.

Таким чином, реалізація цього проєкту продемонструвала ефективність поєднання інструментів для командної роботи, моніторингу та розробки, що забезпечило високий рівень контролю якості та продуктивності. Використання Google Sheets, GitHub та IntelliJ IDEA з Maven дозволило успішно досягти поставлених цілей і створити оптимізовану систему.

3.3 Вдосконалення та генерація зображень за допомогою нейронної мережі

Сучасні технології штучного інтелекту суттєво змінили підхід до роботи з графічним контентом, зокрема до створення, редагування та покращення зображень. Генеративні нейронні мережі відкрили нові можливості в галузі автоматизації художнього процесу, дозволяючи користувачам створювати високоякісні зображення без глибоких знань у сфері комп'ютерної графіки. Завдяки технологіям машинного навчання, особливо дифузійним моделям, стало можливим створювати реалістичні або стилізовані зображення на основі текстових описів або вихідних зразків. Саме тому виникла потреба дослідити, як ці технології можуть бути використані для покращення та генерації зображень.

Серед найпопулярніших сучасних моделей, які активно застосовуються для вирішення таких завдань, особливе місце займає Stable Diffusion. Вона є відкритою для широкого кола розробників та ентузіастів завдяки відкритому вихідному коду. Це сприяє постійному вдосконаленню моделі, оскільки над її розвитком працює велика спільнота, що додає нові функції та оптимізує процес генерації зображень. Завдяки цьому модель активно використовується в багатьох програмних продуктах, забезпечуючи високу ефективність роботи з графічним контентом і значно полегшуючи роботу користувачам. Вона дозволяє автоматизувати процес створення візуального контенту, мінімізуючи потребу у втручанні людини.

Генеративні нейронні мережі — це щось на кшталт творчих машин: вони навчаються на великій кількості прикладів, щоб потім самостійно створювати нові зображення, тексти, музику чи навіть відео. І все це — не просто випадковість, а результат роботи складних алгоритмів, які аналізують шаблони й структури в даних.

Існує кілька видів таких моделей. Наприклад, автоенкодери стискають інформацію, а потім намагаються її відновити — це допомагає їм «зрозуміти» суть зображень. Генеративні змагальні мережі (або GAN-и) — це дві

нейромережі, які змагаються між собою: одна генерує зображення, інша намагається їх розпізнати як фейкові чи справжні. А от дифузійні моделі, до яких належить і добре відома Stable Diffusion, працюють зовсім інакше: вони беруть випадковий шум і поступово, крок за кроком, перетворюють його на зображення. Саме завдяки цьому підходу картинки виходять дуже реалістичними та деталізованими.

Чому дифузійні моделі так популярні? По-перше, вони дають справді високу якість результату — все завдяки тому, що зображення створюється поступово, з увагою до дрібниць. По-друге, вони дуже гнучкі. Можна змінювати стиль, налаштовувати рівень деталізації, додавати художні ефекти — це справжній простір для творчості. Ще один великий плюс — відкритий код. Це означає, що такі моделі можна легко вбудовувати в різні програми або налаштовувати під власні потреби.

І найцікавіше — ці технології вже зараз використовують у найрізноманітніших сферах. Від дизайну і геймдеву до реклами, освіти й медицини. І це тільки початок: чим більше ми з ними працюємо, тим більше бачимо нових можливостей.

Окрім створення нових зображень, нейронні мережі можуть суттєво покращувати вже існуючу графіку. Вони здатні підвищувати роздільну здатність зображень, що є особливо корисним для обробки неякісних фотографій або зображень, зроблених на старих пристроях. Також вони можуть змінювати стиль зображення, імітуючи певні художні техніки або ефекти. Нейронні мережі автоматично коригують кольори та виправляють дефекти, роблячи зображення більш насиченими, яскравими та привабливими. Можливе також відновлення старих або пошкоджених фотографій, що має особливу цінність для архівної роботи та збереження історичних матеріалів. Усі ці можливості значно спрощують процес роботи з графікою, дозволяючи користувачам витрачати менше часу на рутинні завдання.

Для ефективної роботи нейронних мереж у програмному середовищі необхідно правильно організувати їх інтеграцію. У цьому контексті

розробники можуть обирати між локальним розгортанням моделі або використанням зовнішніх API. Якщо модель розгортається локально, наприклад, у випадку зі Stable Diffusion, це дозволяє працювати без підключення до Інтернету та без залежності від сторонніх серверів. Такий підхід забезпечує гнучке налаштування параметрів нейронної мережі та більшу конфіденційність даних. З іншого боку, використання API дозволяє уникнути навантаження на пристрій користувача, оскільки всі обчислення виконуються на сервері, що є особливо важливим для малопотужних пристроїв. Таким чином, вибір залежить від вимог конкретного проєкту та можливостей користувачів.

Stable Diffusion підтримує кілька API, які дозволяють інтегрувати його в різні системи. Основними є API txt2img, який генерує зображення на основі текстових запитів, img2img API, який модифікує існуючі зображення, та inpainting API, який дозволяє замінювати окремі ділянки зображення, зберігаючи загальний стиль і композицію. Крім того, API підтримує можливість роботи з глибокими параметрами налаштування, такими як рівень деталізації, кількість ітерацій та стиль зображення.

Запровадження генеративних нейронних мереж, таких як Stable Diffusion, дозволяє значно спростити та автоматизувати роботу з графікою. Використання цих технологій допомагає скоротити час, необхідний для створення або редагування зображень. Завдяки відкритому коду та активній підтримці спільноти користувачі мають доступ до постійних оновлень та нових можливостей. Оскільки ця технологія постійно вдосконалюється, вона стає ще ефективнішою для різних сфер використання — комерційних проєктів, наукових досліджень або творчих експериментів. У майбутньому очікується ще більше покращень цих моделей, що розширить їх функціональність і зробить процес взаємодії з графічним контентом ще доступнішим та зручнішим.

РОЗДІЛ 4. Практична частина

4.1 Визначення а створення структурних складових програми

Вибір зображення та запуск обробки:

Після того як користувач обирає зображення для обробки, він натискає кнопку «Почати підбір», що запускає алгоритм обробки даних. Вхідними даними є: саме зображення, його розміри, режим роботи машини та значення розширених параметрів (за наявності).

Етап спрощення:

Першим етапом алгоритму є базова спрощена обробка, яка зменшує глибину кольору до 4 бітів. Якщо зображення не є надто складним для друку, програма переходить до фінального етапу. Якщо ж зображення все ще залишається занадто складним для друку, програма ініціює більш детальну обробку.

Комбінація методів спрощення:

Оскільки одноразове спрощення за допомогою одного методу часто не дає задовільної якості, було вирішено застосувати комбіновану стратегію спрощення. Найефективнішими виявилися такі методи:

- кластеризація методом k-means;
- спрощення за принципом найближчого кольору з палітри кольорів (подібної до спектра веселки);
- заміна пікселя на значення сусіднього ліворуч.

Щоб забезпечити перехід між методами, програма використовує ключовий показник — відсоток пікселів, які машина не здатна надрукувати. Якщо цей показник перевищує задані межі, програмне забезпечення автоматично переходить до іншого методу спрощення.

Другий етап спрощення:

Було розроблено палітру з 92 кольорів, яка охоплює діапазон від червоного до фіолетового з рівномірними кроками в системі RGB. До цієї палітри на початку та в кінці масиву додано 7 відтінків сірого та чорного. Для оптимізації обчислень зображення перетворюється в матрицю, де кожен елемент відповідає номеру відповідного кольору в палітрі. Далі обчислюється відсоткова частка кожного кольору в зображенні. Після цього всі кольори, частка яких менша за встановлений поріг «початкового глобального спрощеного узгодження», замінюються на найближчий за відстанню колір із палітри.

Процес спрощення:

Програма переходить до більш детального оброблення, використовуючи значення додаткових параметрів (глобальні межі відбору). На цьому етапі виконується кластеризація методом k-середніх до тих пір, поки відсоток пікселів, непридатних для друку, не досягне заданого рівня. Далі застосовується метод «спрощення до найближчого кольору веселки» до моменту, коли кількість непридатних для друку пікселів досягне нижньої межі відбору. На завершальному етапі програма використовує метод «піксель до пікселя зліва», замінюючи кожен непридатний піксель значенням пікселя ліворуч, аж поки зображення не стане придатним для друку.

Інтеграція нейромережі

Інтеграція нейронної мережі в застосунок передбачає використання API Stable Diffusion у режимі img2img. Це забезпечує трансформацію складних графічних об'єктів із збереженням загального стилю, але зі зниженням рівня деталізації відповідно до вимог обладнання. Нейронна мережа аналізує оригінальне зображення, генерує спрощені версії, адаптовані до палітри машини, та автоматично коригує деталі, які можуть ускладнити друк. Вона згладжує краї, видаляє зайві елементи та покращує загальний вигляд зображення.

Алгоритм роботи нейромережі у програмі

Користувач завантажує зображення та задає параметри обробки. Система аналізує вихідний файл, визначає рівень його складності та передає дані до нейронної мережі. Stable Diffusion створює спрощене зображення, яке потім додатково оптимізується за допомогою алгоритмів кластеризації та зіставлення з палітрою. Користувачеві пропонується кілька варіантів, що дозволяє обрати найбільш придатне рішення для подальшої роботи. Готове зображення експортується у формат, який підтримує машина.

Такий підхід значно скорочує час підготовки графічних даних, мінімізує потребу в ручному редагуванні та підвищує якість кінцевого продукту. Крім того, інтеграція нейронної мережі робить процес адаптації зображення більш інтелектуальним і зручним для користувача, дозволяючи швидше отримувати оптимальні результати без необхідності глибокого налаштування технічних параметрів

Фінальний етап: Після вибору оптимального зображення система виконує фінальну обробку, яка включає адаптацію до технічних параметрів в'язальної машини. Програмне забезпечення конвертує зображення у формат, сумісний із ПЗ машини, враховуючи необхідні вимоги до роздільної здатності, кількості кольорів та щільності пікселів. На цьому етапі також автоматично перевіряється відповідність зображення технологічним можливостям обладнання, що мінімізує ризик браку під час виробництва.

Наступним кроком є перегляд фінальної симуляції візерунка на віртуальній моделі тканини. Це дозволяє оцінити зовнішній вигляд готового виробу до початку виробництва. У разі потреби на цьому етапі можна внести незначні коригування або повернутися до попереднього етапу для покращення параметрів.

Останнім етапом є передача підготовленого файлу до системи керування машиною, після чого розпочинається процес нанесення зображення на в'язаний виріб. Завдяки автоматизованому підходу втручання оператора зведено до мінімуму, що суттєво підвищує ефективність виробничого процесу та скорочує час, витрачений на підготовку файлу.

Такий підхід не лише оптимізує процес підготовки графічних даних, але й забезпечує високу точність і відповідність фінального продукту початковому дизайну. Крім того, інтеграція нейронної мережі робить процес адаптації зображення більш інтелектуальним і зручним для користувача, дозволяючи отримувати оптимальні результати швидше й без необхідності глибоких технічних налаштувань.

4.2 Створення класів з методами

Процес обробки вхідного зображення для створення різних типів фото починається з вибору зображення у форматі 16-бітного BMP або x4 матричного формату. Цей етап є важливим, оскільки вибір формату визначає, як буде оброблятися дані. Формат BMP забезпечує високу якість зображення, що є критично важливим для точності кольору та деталей, а x4 матриця дозволяє зручніше обробляти дані, особливо при роботі з великими зображеннями або взаємодії з іншими системами.

Після завантаження зображення наступним етапом є підбір кольорів. Цей процес є критичним для подальшої роботи, оскільки вибір кольорової палітри впливає на візуальну складову кінцевого продукту. Палітра кольорів повинна бути оптимальною для кожного типу зображення, а також враховувати вимоги до текстури і вигляду готового виробу. Система автоматично вибирає палітри, але також передбачена можливість їх коригування, якщо користувач хоче отримати індивідуальний результат.

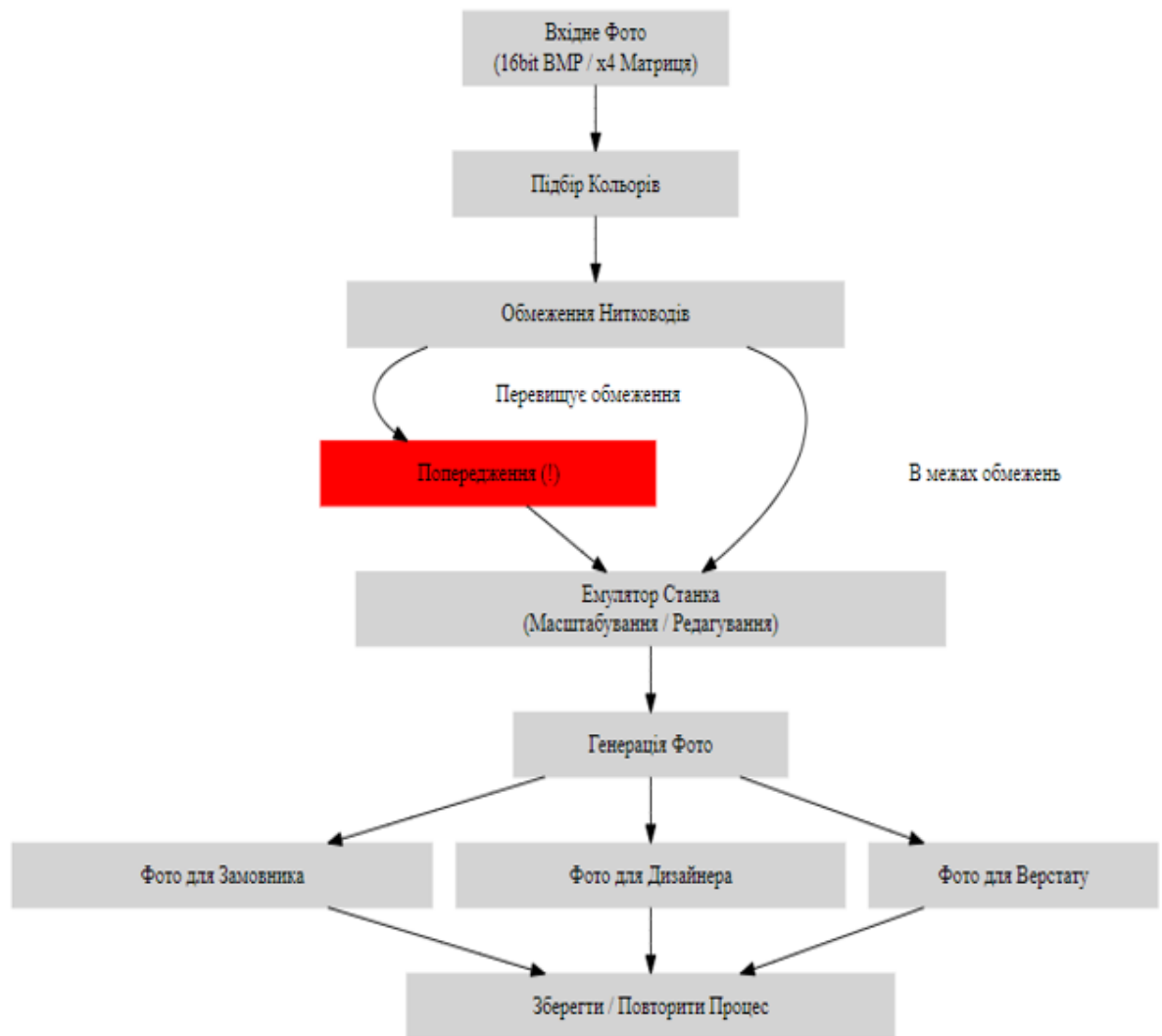
Особлива увага приділяється обмеженням напрямних для ниток, які є ключовими параметрами для роботи з машиною, що виготовляє готову продукцію. Кожен колір і його відтінок мають певні вимоги до товщини нитки, яку використовуватимуть у процесі в'язання або ткацтва. Якщо обрані параметри перевищують допустимі межі для напрямних ниток, система спрацьовує і попереджає користувача про необхідність налаштування параметрів. Це допомагає уникнути помилок, які можуть призвести до поганої якості або неповного виконання замовлення.

Коли всі параметри налаштовані і обмеження дотримано, система переходить до етапу емуляції машини. Тут симулюється реальний процес виготовлення продукції з можливістю масштабування зображення та редагування його для точнішого відображення кінцевого результату. На цьому етапі користувач може перевірити, як виглядатиме готове зображення в реальних умовах машини, що є особливо важливим для тих, хто працює зі складними або нестандартними дизайнами.

Генерація зображень завершується створенням трьох версій результату, кожна з яких призначена для конкретних цілей: одна для клієнта для відображення готового дизайну в високій якості, інша для дизайнера для внесення необхідних коригувань, і третя для безпосереднього використання на машині, що вже враховує всі технічні обмеження та специфікації.

Останнім кроком у процесі є можливість зберегти результати або повернутися до попередніх етапів для налаштування параметрів, що дозволяє гнучко керувати процесом розробки та досягати найкращих результатів. Це дозволяє користувачам або розробникам зберігати поточний стан проєкту на будь-якому етапі, даючи можливість безпечно експериментувати з різними параметрами без ризику втратити вже виконану роботу. Повернення до попередніх етапів особливо корисне, коли потрібно внести коригування або протестувати нові налаштування без необхідності починати процес спочатку. Ця гнучкість дозволяє налаштувати процес відповідно до індивідуальних вимог користувача або системи, забезпечуючи максимальну кастомізацію та можливість точно налаштувати результат під конкретні умови. Завдяки цій функціональності користувачі можуть досягти ідеального результату шляхом численних ітерацій та тестування різних варіантів, поки не знайдуть найбільш ефективне рішення для своїх потреб. Це також забезпечує більший контроль над процесом і дозволяє ефективно реагувати на змінювані вимоги або нові дані, що з'являються в процесі роботи.

Таким чином, блок-схема демонструє не лише етапи обробки зображень, а й гнучкість системи, яка дозволяє коригувати параметри на кожному етапі, забезпечуючи максимальну точність і якість кінцевого продукту.



4.3 Діаграма роботи програми

Блок-схема зображена на рисунку 4.1 надає візуальне розуміння процесу обробки і генерації зображень в інформаційній системі, яку можна

використовувати в автомобільному виробництві для інтеграції з автоматизованими системами контролю якості або дизайну продукції.

Рисунок 4.1 Блок схема роботи програми

Основний метод цього процесу полягає в перевірці друкованості конкретного зображення. Програмне забезпечення отримує спрощену фотографію і аналізує її, щоб визначити, чи відповідає вона вимогам для друку. Залежно від обраного режиму, додаток або повертає об'єкт, що містить інформацію про кольори та карту для друкарської машини, або повідомляє, що фото не підлягає друку. Зокрема, якщо зображення можна надрукувати, об'єкт включає детальну інформацію про колір нитки, яку потрібно протягнути в кожен з напрямних ниток для точного відтворення зображення. Карта для машини визначає, як саме шари кольорів накладаються один на один. Якщо аналіз показує, що зображення не відповідає технічним вимогам, програмне забезпечення повертає інформацію про неможливість друку.

Один з ключових елементів системи — це об'єкт, що відповідає за зберігання детальної інформації про зображення, яке буде оброблятися машиною. Цей об'єкт містить наступні основні параметри:

Розмір зображення – інформація про ширину та висоту в пікселях, що є важливою для визначення масштабу та позиціонування зображення на матеріалі.

Колір кожного пікселя – матриця або карта, яка відображає кольори всіх пікселів зображення. Це дозволяє машині відтворювати зображення з високою точністю, враховуючи характеристики матеріалів, з якими працюють.

Мапа для верстату – Специфічні інструкції для керування процесом обробки: шлях різця, порядок нанесення кольору, швидкість обробки та параметри глибини обробки. Ця карта дозволяє налаштувати параметри машини для відтворення зображення відповідно до технічних вимог та характеристик оброблюваного матеріалу.

```

public class photo {
    private Color[][] pixels;

    private int[][] StanokScheme;

    private int n;

    private int m;

    public photo(int n, int m) {

        this.n = n;

        this.m = m;

        this.pixels = new Color[n][m];

        this.StanokScheme = new int[6][3];

    }

```

У визначенні структури застосунку для оптимізації комп'ютерних ресурсів і прискорення обробки фото перетворюється на числову матрицю. Кожному пікселю зображення присвоюється числове значення, що представляє його колір або яскравість, і це значення зберігається у двовимірній або тривимірній матриці, залежно від кольорової моделі (наприклад, RGB або відтінки сірого).

Цей підхід дозволяє ефективно виконувати математичні операції над зображенням, такі як зміна яскравості або застосування фільтрів, що знижує обчислювальні витрати. Замість складних маніпуляцій з графічними форматами, обробка числових даних значно прискорює роботу додатку та зменшує обсяг пам'яті, необхідної для роботи з великими зображеннями. Це досягається за допомогою наступного методу:

```

public static int[][] convertToColorIndices(photo photoObject) {
    int n = photoObject.getN();
    int m = photoObject.getM();
    int[][] colorIndices = new int[n][m];

```

```

for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        Color pixelColor = photoObject.getPixel(i, j);
        int colorIndex = findColorIndex(pixelColor);
        colorIndices[i][j] = colorIndex;
    }
}

return colorIndices;
}

```

4.4 Тестування функціональності готового продукту

4.5

Після створення початкової версії програмного забезпечення було проведено етап компіляції та тестування, під час якого були виявлені перші недоліки та проблеми, які вимагали негайного вирішення. Після того, як програмне забезпечення було передано користувачам для огляду та отримання відгуків, було виявлено низку важливих аспектів, які потребували подальшого розвитку. На цьому етапі були проаналізовані отримані коментарі, і були внесені необхідні корективи у функціональність та інтерфейс додатку.

Однією з основних проблем, що виникли під час першого етапу тестування, було некоректне налаштування параметрів обробки зображень. Спочатку передбачалося, що розширені налаштування будуть підходити для одного шаблону фотографії. Однак через різноманіття умов зйомки, освітлення та якості зображень виникла необхідність індивідуального підбору параметрів для кожної фотографії. Це вимагало змін у програмному забезпеченні, щоб налаштування були адаптовані до кожного конкретного випадку. В результаті було прийнято рішення перемістити ці параметри в

окремий блок налаштувань, що значно полегшило налаштування додатку для різних зображень і забезпечило більшу гнучкість у використанні.

Іншою важливою частиною ревізії було усунення багів, пов'язаних з некоректною роботою програми при обробці заблокованих напрямних ниток. Ці баги виникали, коли додаток не міг правильно обробляти або записувати зміни в нитках, що знаходилися в заблокованому стані, що призводило до нестабільної роботи. Було проведено глибокий аналіз внутрішньої логіки обробки ниток, і були внесені зміни в алгоритми, що дозволило коректно обробляти всі випадки заблокованих ниток.

Іншим важливим кроком була робота з інтерфейсом програмного забезпечення. Під час тестування з'ясувалося, що початкове розуміння термінології в'язання було недостатньо точним, що призводило до плутанини з термінами, такими як «петлі» та «стовпці». З урахуванням цього термінологію було уточнено, а описи кожної функції були вдосконалені для того, щоб користувачам було легше орієнтуватися в можливостях програми. Цей крок був важливий для поліпшення зручності користування додатком і зниження ймовірності помилок під час його використання.

Крім того, необхідність зберігання фотографій стала важливим аспектом. На першому етапі передбачалося, що всі три фотографії зберігатимуться одночасно, але на практиці з'ясувалося, що користувачам зручніше зберігати кожену фотографію окремо. Це зменшило навантаження на систему зберігання та значно спростило доступ до кожного окремого зображення.

Останній етап вдосконалення додатку включав перевірку сумісності з іншими системами та забезпечення безпечної роботи в умовах великої кількості одночасних користувачів. Для цього було впроваджено кілька додаткових заходів із захисту даних і покращення продуктивності, що дозволили додатку стабільно функціонувати навіть за високих навантажень.

В результаті вдосконалень програмне забезпечення повністю відповідає вимогам користувачів, забезпечуючи високу ефективність і стабільність при обробці зображень для в'язання шкарпеток. Програмне забезпечення було

покращено шляхом удосконалення можливостей налаштування, що дозволяє користувачам точніше адаптувати систему до індивідуальних потреб та специфікацій їх виробничих процесів. Зокрема, алгоритми обробки зображень були оптимізовані, що значно знижує час обробки зображень і покращує точність та якість результатів. Також були виправлені численні баги в інтерфейсі, що зробило програмне забезпечення більш зручним і інтуїтивно зрозумілим.

Працюючи над вдосконаленням алгоритмів, ми зробили їх більш надійними та ефективними, що зменшує ймовірність помилок у процесі в'язання та забезпечує більш стабільні результати. Зокрема, нам вдалося вирішити проблеми, пов'язані з некоректною обробкою певних типів зображень, а також покращити взаємодію з користувачем, так що тепер можна легко налаштувати всі необхідні параметри без необхідності звертатися до технічної документації чи допомоги. Це значно підвищує комфорт використання додатку і дозволяє користувачам зосередитися на основних завданнях без зайвих втручань.

В результаті розробки та вдосконалення програмного забезпечення для обробки зображень для в'язання шкарпеток були успішно усунуті виявлені недоліки та баги. Налаштування параметрів були оптимізовані, виправлені помилки в обробці заблокованих напрямних ниток та уточнено термінологію. Функція збереження зображень також була вдосконалена, що забезпечило більшу зручність для користувачів. В результаті програма стала більш стабільною, гнучкою і відповідає вимогам кінцевих користувачів, що підтверджує досягнення поставлених цілей. Приклади роботи додатку наведені на малюнках 4.1 та 4.2.

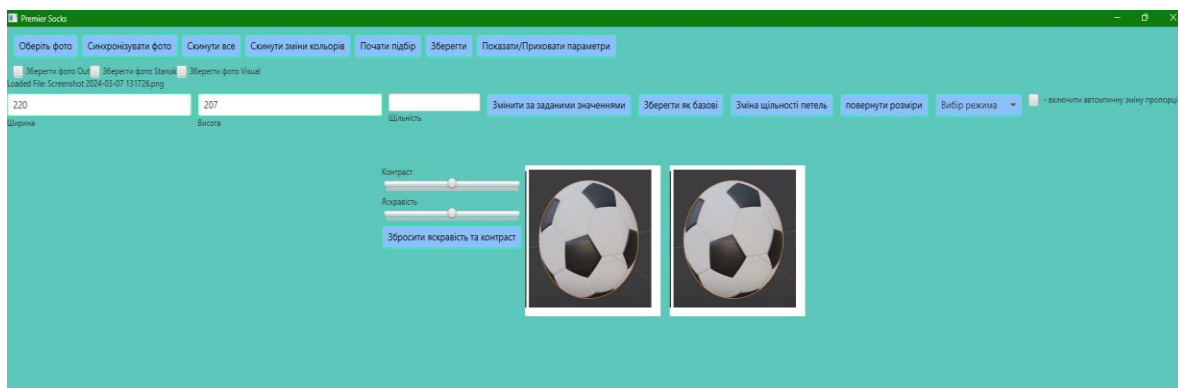


Рисунок 4.2 Приклад інтерфейсу під час роботи

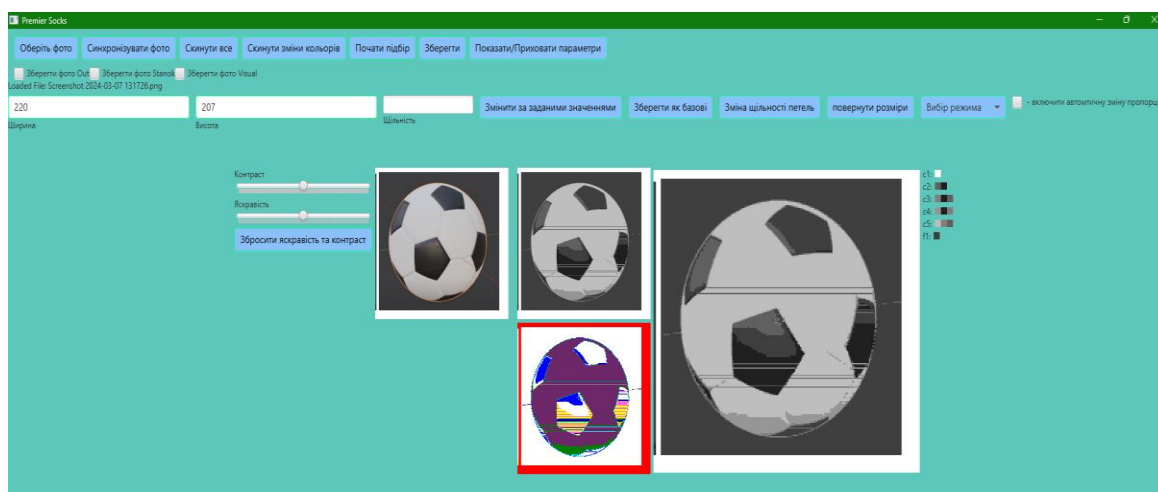


Рисунок 4.3 Вихідний результат роботи програми

Готова програма пройшла всі етапи тестування і була остаточно затверджена. Після цього вона була передана замовнику, який ретельно перевіряв її функціональність, відповідність технічному завданню та інтеграцію з іншими виробничими системами. Остаточна версія повністю відповідала вимогам замовника. Після впровадження програми в виробничий процес, вона стала невід'ємною частиною автоматизованого управління виробництвом шкарпеток, забезпечуючи оптимізацію ресурсів, значні підвищення ефективності та усунення людських помилок на кожному етапі проектування.

4.5 Алгоритм роботи нейронної мережі

У процесі розробки програмного забезпечення однією з ключових задач стало налагодження взаємодії з нейронною мережею Stable Diffusion.

Основна мета — автоматично спрощувати зображення для подальшого використання на трикотажному верстаті. Ідея полягала в тому, щоб користувач міг завантажити довільне зображення, вказати необхідні параметри (розміри, палітру, режим генерації тощо), а далі вся обробка проходила б із мінімальним ручним втручанням.

Після вибору зображення система спочатку проводить його оцінку — визначає складність, кількість кольорів, наскільки воно підходить для друку на обладнанні. Якщо все гаразд, картинка передається до Stable Diffusion через API (в режимі `img2img`), де генерується нова версія — менш деталізована, але зі збереженням загального стилю й форми. Далі це зображення проходить ще один етап локальної обробки. Зокрема, використовується кластеризація (наприклад, метод `k-means`), яка дозволяє зменшити кількість «важких» для друку пікселів. Також оптимізується палітра, щоби кольори відповідали технічним можливостям верстата.

Якщо система виявляє, що зображення все ще містить занадто багато непридатних кольорів, автоматично застосовуються додаткові спрощення: наприклад, заміна проблемних кольорів на найближчі допустимі. Потім користувачеві показується кілька варіантів на вибір, і він може самостійно обрати найкращий. Після цього вибране зображення адаптується під конкретні параметри верстата: враховується роздільна здатність, формат, кількість кольорів та інші технічні обмеження. Перед фінальним експортом система ще раз перевіряє, чи все відповідає вимогам. Це важливо, бо навіть незначне відхилення може вплинути на якість готового виробу.

Такий підхід дозволяє значно спростити й прискорити підготовку графіки для друку. Оператору вже не потрібно вручну редагувати зображення чи перевіряти відповідність — усе виконується автоматично. Крім того, впровадження нейромережі дало змогу не просто автоматизувати рутинні дії, а й підняти загальний рівень точності та стабільності. У підсумку — менше браку, менше витрат часу і ресурсів, а також вищий рівень зручності для користувача. На мій погляд, саме завдяки поетапному підходу — від аналізу

зображення до його генерації та оптимізації — вдалося створити інструмент, який реально підходить для використання в умовах текстильного виробництва. Він не тільки полегшує роботу операторів, а й відкриває можливості для інтеграції новітніх технологій у традиційні виробничі процеси.

4.6 Додавання методів для взаємодії з нейромережею

У межах дипломного проєкту були розроблені та реалізовані методи взаємодії з нейронною мережею Stable Diffusion через API AUTOMATIC1111 WebUI. Це дає змогу автоматизувати процес генерації та модифікації зображень на основі текстових запитів або вхідних зображень. Основна мета цієї розробки — створити ефективний інструмент для обробки графічних даних, який може бути використаний у різних галузях, таких як дизайн, текстильна промисловість, цифрове мистецтво та автоматизоване створення ілюстрацій.

Для реалізації взаємодії було створено клас **StableDiffusionClient**, який містить набір функцій для генерації зображень та їх обробки за допомогою нейронної мережі. Зокрема, клас надає метод для створення зображення на основі текстового опису. Це дає змогу користувачу, який не має навичок роботи з графічними редакторами, отримати якісні ілюстрації за лічені секунди. Це реалізовано за допомогою наступного методу:

```
public Image generateImage(String prompt) throws IOException {
    return generateImage(prompt, new Parameters());
}
```

Якщо потрібно детальніше описати процес генерації, можна скористатися методом, який дозволяє передавати додаткові параметри. Це відкриває можливість тонкого налаштування зображення, зокрема вибору стилю, рівня деталізації та інших параметрів генератора нейронної мережі.

```
public Image generateImage(String prompt, Parameters params) throws
IOException {
```

```

JSONObject requestBody = params.toTextToImageJson(prompt);
String endpoint = this.apiUrl + "/txt2img";
String response = sendPostRequest(endpoint, requestBody.toString());
JSONObject responseJson = new JSONObject(response);
String imageBase64 = responseJson.getJSONArray("images").getString(0);
byte[] imageBytes = Base64.getDecoder().decode(imageBase64);
return ImageIO.read(new ByteArrayInputStream(imageBytes));
}

```

Окрім створення нових зображень, клас також надає можливість обробляти наявні. Це особливо корисно в ситуаціях, коли потрібно змінити зображення, покращити його якість або надати йому нового стилю. Наприклад, можна перетворити фотографію на картину, стилізовану під певного художника. Це реалізується за допомогою наступного методу:

```

public Image processImage(Image inputImage, String prompt) throws
IOException {
    return processImage(inputImage, prompt, new Parameters());
}

```

Для більшої гнучкості доступна розширена версія цього методу, яка дозволяє передавати додаткові параметри, що впливають на кінцевий результат. Зокрема, можна налаштувати ступінь впливу нейронної мережі на оригінальне зображення, визначити рівень деталізації та інші важливі характеристики.

```

public Image processImage(Image inputImage, String prompt, Parameters
params) throws IOException {
    String imageBase64 = imageToBase64(inputImage);
    JSONObject requestBody = params.toImageToImageJson(prompt,
imageBase64);
    String endpoint = this.apiUrl + "/img2img";
    String response = sendPostRequest(endpoint, requestBody.toString());
    JSONObject responseJson = new JSONObject(response);
}

```

```

        String                                resultImageBase64                =
responseJson.getJSONArray("images").getString(0);
        byte[] imageBytes = Base64.getDecoder().decode(resultImageBase64);
        return ImageIO.read(new ByteArrayInputStream(imageBytes));
    }

```

Ще однією важливою функцією є можливість збереження параметрів генерації у файл формату JSON. Це дозволяє повторно використовувати налаштування, що особливо корисно при створенні серії зображень або під час повторення експериментів із різними параметрами. Збережені файли можна використовувати для аналізу та порівняння результатів генерації, а також для автоматизації процесу роботи з нейронною мережею. Це реалізується за допомогою наступного методу:

```

public void saveParamsToJsonFile(String filePath, String prompt, Parameters
params, boolean isImageToImage) throws IOException {
    JSONObject json;
    if (isImageToImage) {
        json = params.toImageToImageJson(prompt, "");
        json.remove("init_images");
        json.put("using_init_image", true);
    } else {
        json = params.toTextToImageJson(prompt);
    }
    json.put("generation_type", isImageToImage ? "img2img" : "txt2img");
    json.put("timestamp", System.currentTimeMillis());
    Files.write(new                                     File(filePath).toPath(),
json.toString(2).getBytes(StandardCharsets.UTF_8));
}

```

Дані обмінюються з сервером через HTTP-запити, що є стандартом для комунікації між клієнтською частиною програми та сервером. Такий підхід дозволяє ефективно інтегрувати програму з API нейронної мережі,

забезпечуючи швидку передачу запитів та відповідей, а також можливість динамічного обміну даними без значних затримок. Використання HTTP-запитів є оптимальним для роботи з різними типами даних, оскільки цей протокол підтримує великі обсяги інформації та може бути легко масштабований під різні вимоги. Це особливо важливо при роботі з великими зображеннями, де необхідна висока швидкість обробки для забезпечення безперервної роботи та уникнення затримок.

Крім того, при обробці серії запитів в реальному часі використання HTTP-запитів дозволяє паралельну обробку запитів, що зменшує час очікування та підвищує загальну продуктивність системи. Така архітектура не лише забезпечує ефективний обмін даними, але й робить систему більш адаптивною до змін навантаження, що є важливим при роботі з великими обсягами даних або з вимогами до високої швидкості обробки. Завдяки цьому користувачі можуть отримувати миттєві відповіді на свої запити, що значно підвищує зручність і ефективність роботи з програмою.

Реалізовані методи забезпечують ефективну та гнучку взаємодію з нейронною мережею, що значно спрощує інтеграцію генеративних можливостей у розроблену програмну систему. Вони дозволяють створювати високоякісні графічні матеріали за мінімальний час, що робить їх корисними для широкого кола користувачів — від дизайнерів до інженерів, які працюють з текстильними та графічними процесами.

Розробка та впровадження методів взаємодії з нейронною мережею Stable Diffusion створило потужний та гнучкий інструмент для генерації та обробки зображень. Завдяки можливості налаштовувати параметри генерації, зберігати конфігурації у JSON-файлах та використовувати HTTP-запити для взаємодії з API, система забезпечує ефективну та автоматизовану роботу з графічними даними. Ця технологія відкриває нові можливості в цифровому мистецтві, текстильній промисловості та інших сферах, де важливо швидко отримувати високоякісні зображення на основі текстових описів або модифікації наявних зображень. Подальший розвиток проєкту може включати оптимізацію

продуктивності, підтримку нових моделей нейронних мереж та розширену функціональність для роботи з різними графічними форматами.

ВИСНОВКИ

У процесі виконання дипломного проекту було розроблено комплексне програмне рішення для обробки та адаптації зображень, призначених для виробництва в'язальних виробів. Основною метою було створити ефективний алгоритм обробки, який спрощував би зображення та оптимізував їх для використання на спеціалізованих машинах. Дослідження та впровадження проекту призвели до значних результатів в автоматизації процесу підготовки зображень, що є важливим для текстильної промисловості.

Було проведено детальний аналіз існуючих методів кластеризації та спрощення зображень, серед яких алгоритм k-means та методи зменшення кольорових палітр виявилися найбільш ефективними. Вибір цих підходів був обґрунтований необхідністю мінімізації кількості кольорів, що використовуються у виробництві в'язальних виробів, а також для покращення чіткості отриманого малюнка. Було проведено низку експериментальних досліджень, щоб підтвердити високу ефективність обраного підходу, що значно знижує візуальний шум та спрощує форми без втрати основних деталей зображення.

На основі досліджень була розроблена програмна реалізація, яка дозволяє користувачам виконувати автоматичну обробку зображень через зручний інтерфейс. Програмне забезпечення підтримує гнучке налаштування параметрів обробки, що дає змогу використовувати його для різних типів зображень та текстильних виробів. Процес тестування підтвердив правильність алгоритмів, що забезпечують високу якість спрощення графіки без значних втрат у деталях. Результати тестів також підтвердили стабільність програмного рішення за різних параметрів вхідних зображень.

Крім того, було проведено порівняльний аналіз продуктивності розробленого програмного забезпечення з аналогічними існуючими рішеннями, який показав його конкурентоспроможність за швидкістю обробки, точністю адаптації зображень і ступенем автоматизації процесу.

Додатково до розробленого програмного забезпечення була інтегрована штучна нейронна мережа для адаптації зображень до специфічних параметрів машин, що відкриває можливість подальшого вдосконалення системи. Це робить обробку більш гнучкою та адаптивною до потреб виробничого процесу. Використання нейронної мережі також пропонує перспективи для автоматичного навчання системи на основі зібраних даних, що забезпечить поступове покращення якості обробки з накопиченням досвіду.

Подальше розширення функціональності системи може включати використання технологій глибокого навчання для оптимізації параметрів зображень у реальному часі, що забезпечить ще більш точне відтворення складних візерунків на в'язаних виробах.

Розроблена система є перспективним рішенням для текстильної промисловості, оскільки дозволяє автоматизувати рутинні процеси підготовки зображень, збільшуючи ефективність виробництва та знижуючи ризик помилок. Запропоноване рішення допомагає скоротити час, витрачений на налаштування параметрів зображень, що є критичним для масового виробництва в'язаних виробів.

Подальший розвиток системи може включати вдосконалення алгоритму нейронної мережі, що дозволить ще точніше адаптувати зображення до технологічних можливостей машин, а також розширення функціональності програмного забезпечення, зокрема інтеграцію з CAD/CAM системами та можливість автоматичного виправлення помилок під час малювання.

Загалом, результати дослідження показують, що використання сучасних алгоритмів обробки зображень у поєднанні з нейронними мережами є перспективним напрямком для автоматизації текстильного виробництва. Отримані розробки можуть бути впроваджені у виробничі процеси та адаптовані до потреб різних типів в'язальних машин, що сприятиме підвищенню конкурентоспроможності підприємств у цій галузі.

У майбутньому, з урахуванням розвитку комп'ютерного зору та технологій глибокого навчання, можна подальше розширення можливостей

системи, включаючи автоматичне розпізнавання візерунків, оптимізацію розташування візерунків на тканині та поліпшену адаптацію до різних технологічних вимог. Це відкриває нові горизонти для автоматизації та цифровізації текстильного виробництва, що є ключовим фактором у сучасному конкурентному середовищі.

Крім того, запропоноване рішення має значний потенціал для розширення поза межі текстильної промисловості. Систему можна адаптувати для використання в інших галузях, таких як виробництво кераміки, гравірування металу та дерева, а також виготовлення друкованої продукції. Це демонструє універсальність розробленої методології та можливість її масштабування для різних технологічних процесів.

Подальший розвиток досліджень у цій сфері може включати аналіз можливостей інтеграції штучного інтелекту для створення повністю автоматизованих систем, які самостійно вибирають параметри обробки та адаптують зображення до специфічних вимог виробництва.

Розробка та впровадження цієї системи дозволяє компаніям не лише оптимізувати процеси підготовки зображень, але й значно скоротити виробничий цикл, що є особливо важливим для великих текстильних компаній та стартапів, які прагнуть швидко вийти на ринок з новими продуктами. У майбутньому подальше впровадження сучасних технологій, таких як штучний інтелект, автоматизований аналіз та інтелектуальні системи контролю якості, дозволить текстильній промисловості досягти нового рівня ефективності.

СПИСОК ЛІТЕРАТУРИ

1. Морато, К. Р. *JavaFX 17 by Example*. Apress. (2021). Ст видавництва
2. Вівер, Дж., Ченг, В., Селлер, Л., та ін. (2021). *Pro JavaFX 9* (оновлене видання). Apress.
3. Лоу, Д. (2022). *JavaFX for Dummies* (оновлене видання). For Dummies.
4. Блох, Дж. (2023). *Modern JavaFX Development*. O'Reilly Media.
5. Coding Tech Room (2024) *Implementing an AI-Based Image Generator in Java*.
6. k-means clustering [Електронний ресурс] / Матеріал з Вікіпедії — вільної енциклопедії. – Режим доступу: https://en.wikipedia.org/wiki/K-means_clustering
7. Томпсон, Дж. (2019). *Apache Maven: Beginner to Guru*. Компанія Sonatype. (2019). *Maven: The Definitive Guide*.
8. Ольховська О. Методичні рекомендації щодо оформлення пояснювальних записок до курсових проектів (робіт) для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки та інформаційні технології», «Комп'ютерні науки» галузь знань - 12 «Інформаційні технології»
9. Лов, Д. *JavaFX for Dummies (Updated Edition)*. For Dummies, (2022).
10. Ромбач, Р., Блатман, А., Лоренс, Д., Ессер, П., & Оммер, Б. (2022). *High Resolution Image Synthesis with Latent Diffusion Models*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [Електронний ресурс] / Матеріал з Архів — безкоштовний сервіс розповсюдження та архів з відкритим доступом. – Режим доступу: <https://arxiv.org/abs/2112.10752>
11. Гудфеллов, І., Бенгіо, Й., & Коурвіл, А. (2016). *Deep Learning*. MIT Press.
12. Гонзалес, Р., Вудс Р., (2018). *Digital Image Processing (4th Edition)*. Pearson.