
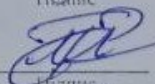



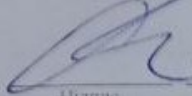
Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Спосіб віртуальної примірки одягу з використанням генеративних нейронних мереж

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 4 курсу, група КН-19-1  М.С. Оксаниук
Курс, група виконавця Ініціали, прізвище
Керівник: док. філ., ст. викладач кафедри КН  П.М. Радюк
Науковий ступінь, посада Ініціали, прізвище
Нормоконтроль: к.т.н., доцент кафедри КН  Р.О. Багрії
Науковий ступінь, посада Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор  О.В. Бармак
Ініціали Ініціали, прізвище

01 06 2023 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь бакалавр

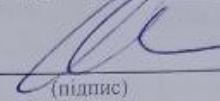
Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

Освітня програма освітньо-професійна програма підготовки бакалавра

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук



(підпис)

д.т.н., професор О.В. Бармак

«06» 03 2023 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Спосіб віртуальної примірки одягу з використанням генеративних нейронних мереж»

2. Завдання видано студенту Оксанюку Максиму Сергійовичу
(прізвище, ім'я, по батькові)

3. Керівник роботи ст. викладач кафедри КН Радюк Павло Михайлович
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від «01» 03 2023р. № 5

5. Дата видачі завдання студенту: «03» 03 2023р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Провести аналіз засобів штучного інтелекту для оброблення та генерування цифрових зображень та обрати найкращий; використати обраний засіб оброблення та генерування цифрових зображень для розв'язання задачі віртуальної примірки за цифровими зображеннями людини та предметів одягу; реалізувати спосіб віртуальної примірки одягу у вигляді модуля програмного забезпечення; провести експериментальне тестування реалізованого модуля за еталонними наборами даних; вихідними даними модуля є нове зображення людини із накладеним зображенням предмету одягу.

7. Календарний план виконання кваліфікаційної роботи бакалавра:


№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником	грудень 2022	виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2023	виконано
3	Робота над розділом 1 – Сучасний стан віртуальної примірки одягу за цифровими зображеннями людини та предметів одягу	січень 2023	виконано
4	Робота над розділом 2 – Спосіб віртуальної примірки одягу з використанням генеративних нейронних мереж	березень 2023	виконано
5	Робота над розділом 3 – Програмна реалізація інформаційної системи з використанням генеративної нейронної мережі	квітень 2023	виконано
6	Оформлення пояснювальної записки згідно вимог	травень 2023	виконано
7	Попередній захист кваліфікаційної роботи бакалавра	травень 2023	виконано
8	Захист кваліфікаційної роботи бакалавра на засіданні Екзаменаційної комісії	червень 2023	виконано

Виконавець: студент 4 курсу, група КН-19-1
Курс, група виконавця


Підпис

М.С. Оксанюк
Ініціали, прізвище

Керівник: док. філ., ст. викладач кафедри КН
Науковий ступінь, посада


Підпис

П.М. Радюк
Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: Спосіб віртуальної примірки одягу з використанням генеративних нейронних мереж

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-19-1 Оксанюк Максим Сергійович

Керівник кваліфікаційної роботи бакалавра: доктор філософії, ст. викладач кафедри КН Радюк Павло Михайлович

Кваліфікаційна робота бакалавра містить:


Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
65	30	2	22	2

Метою кваліфікаційної роботи бакалавра є покращення віртуальної примірки одягу за цифровими зображеннями людини та предметів одягу.

Зміст пояснювальної записки (перелік задач) та вихідні дані: провести аналіз засобів штучного інтелекту для генерування цифрових зображень та обрати найкращий; використати обраний засіб оброблення та генерування цифрових зображень; реалізувати спосіб віртуальної примірки одягу у вигляді модуля програмного забезпечення; провести експериментальне тестування реалізованого модуля за еталонними наборами даних; вихідними даними модуля є нове зображення людини із накладеним зображенням предмету одягу.

Досягнення мети кваліфікаційної роботи бакалавра полягає у створенні інформаційної системи, використання якої дасть змогу покращити процес віртуальної примірки одягу за цифровими зображеннями людини та предметів одягу.

Ключові слова: індустрія моди, віртуальна примірка, цифрове зображення, аналіз зображення, генеративна нейронна мережа.

Виконавець: студент 4 курсу, група КН-19-1  М.С. Оксанюк
Курс, група виконавця Підпис Ініціали, прізвище

Зміст

Перелік скорочень	3
Вступ.....	4
Розділ 1 Сучасний стан віртуальної примірки одягу за цифровими зображеннями людини та предметів одягу	6
1.1 Огляд використання інформаційних технологій в індустрії моди	6
1.2 Аналіз методів, засобів та технологій оброблення та генерування цифрових зображень	9
1.3 Використання нейронних мереж для віртуальної примірки одягу.....	12
1.4 Мета, завдання та вимоги до реалізації інформаційної системи	16
Розділ 2 Спосіб віртуальної примірки одягу з використанням генеративних нейронних мереж.....	18
2.1 Спосіб формалізації та оброблення цифрового зображення	18
2.2 Проектування архітектури генеративної нейронної мережі для віртуальної примірки одягу	26
2.3 Функціональна структура інформаційної системи на основі генеративної нейронної мережі	29
2.4 Проектування структури інформаційної системи	33
2.4.1 Інфологічна модель даних.....	33
2.4.2 Проектування інтерфейсу інформаційної системи	37
2.5 Висновки до розділу 2	39
Розділ 3 Програмна реалізація інформаційної системи з використанням генеративної нейронної мережі	41
3.1 Структура та функціональне призначення програмних складових інформаційної системи	41
3.2 Особливості реалізації програмних складових інформаційної системи	50
3.3 Експериментальне тестування інформаційної системи	54
3.4 Вимоги до розгортання інформаційної системи та інструкція користувача.....	57
3.5 Висновки до розділу 3	61
Висновки	62
Перелік посилань.....	63
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
CAD	Computer-Aided Design
CNN	Convolutional neural network
GNN	Graph Neural Network
GAN	Generative Adversarial Networks
GUI	Graphic User Interface
GMM	Gaussian Mixture Models
DCGAN	Deep Convolutional GAN
CGAN	Conditional GAN
RGB	Red, Green, Blue
JPEG	Joint Photographic Experts Group
PNG	Portable Network Graphics
GIF	Graphics Interchange Format
BMP	Device independent bitmap
TIFF	Tag Image File Format
TPS	Temporary Protected Status
ГНМ	Генеративно нейронні мережі
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки

Вступ

Кваліфікаційна робота бакалавра присвячена розв'язанню задачі віртуальної примірки одягу з використанням генеративних нейронних мереж.

Актуальність. Віртуальна примірка одягу за допомогою цифрових зображень набуває все більшої популярності та значущості в сучасному світі. Застосування штучного інтелекту для генерації цифрових зображень та обробки даних дозволяє створювати реалістичні та точні віртуальні моделі одягу, що має великий потенціал у різних сферах, таких як мода, електронна комерція, дизайн одягу та інше.

Створення інформаційної системи, яка забезпечує віртуальну примірку одягу, дозволить користувачам зручно та точно оцінювати вигляд різних предметів одягу на собі без необхідності фізичного приміряння. Це може полегшити процес вибору одягу, зекономити час та зусилля, а також допомогти виробникам та дизайнерам у покращенні своїх продуктів та послуг.

Таким чином, розробка інформаційної системи для віртуальної примірки одягу є актуальною темою, яка поєднує передові технології штучного інтелекту, комп'ютерного зору та моделювання, впливає на розвиток модної індустрії та сприяє зручності та ефективності в процесі вибору та покупки одягу.

Об'єкт дослідження – процес примірки одягу за цифровими зображеннями людини та предметів одягу.

Предмет дослідження – методи, засоби та технології оброблення та генерування цифрових зображень.

Мета кваліфікаційної роботи бакалавра – покращення віртуальної примірки одягу за цифровими зображеннями людини та предметів одягу.

Завдання кваліфікаційної роботи бакалавра – для досягнення поставленої мети визначено наступні завдання:

1. Провести аналіз засобів штучного інтелекту для оброблення та генерування цифрових зображень та обрати найкращий.

2. Використати обраний засіб оброблення та генерування цифрових зображень для розв'язання задачі віртуальної примірки за цифровими зображеннями людини та предметів одягу.

3. Реалізувати спосіб віртуальної примірки одягу у вигляді модуля програмного забезпечення.

4. Провести експериментальне тестування реалізованого модуля за еталонними наборами даних.

Розділ 1 Сучасний стан віртуальної примірки одягу за цифровими зображеннями людини та предметів одягу

1.1 Огляд використання інформаційних технологій в індустрії моди

Індустрія моди – це всесвітній ринок із багатьма захоплюючими тенденціями та модою. Це дуже динамічна галузь, яка постійно змінюється, для того щоб задовольнити потреби своїх клієнтів [1]. Щоб бути в курсі останніх тенденцій, модельєри активно використовують інформаційні технології. Ці технології допомагають дизайнерам створювати інноваційні та нові проекти [2]. Вони також використовують ці технології для прототипування своїх дизайнів і перевірки ефективності своїх творінь.

Ось кілька прикладів використання інформаційних технологій в даній індустрії:

1. CAD (Computer-Aided Design) [3] – це технологія, яка створює та моделює об'єкти у віртуальному середовищі за допомогою спеціального програмного забезпечення. Різні CAD-програми дозволяють інженерам, архітекторам та дизайнерам створювати, редагувати та оптимізувати різноманітні проекти у 2D або 3D форматі.

У модній індустрії, CAD-програми використовуються для розробки шаблонів, створення дизайну, вирішення конструктивних завдань та роботи з кроєм.

У загальному, ці програми є важливим інструментом для багатьох дизайнерських та інженерних проектів, які вимагають великої точності та складної геометрії. Вони дозволяють значно зменшити витрати та заощадити час на проектуванні, зробити процес більш точним та ефективним [4].

2. Виробництво. У ньому застосовуються технології автоматизації, такі як роботи та комп'ютерно-чутливі машини, що допомагають підвищити продуктивність і знизити витрати на виробництво матеріалу.

3. Електронна комерція. Інтернет-магазини стали дуже популярними серед споживачів. Вони дозволяють покупцям швидко та зручно здійснювати покупки, а також порівнювати ціни та ознайомлюватися з відгуками про різні товари.

4. Віртуальна реальність. Технологія віртуальної реальності дозволяє дизайнерам створювати віртуальні приміщення для показів мод та примірки одягу [5], що зменшує витрати на реалізацію колекцій та кількість виробничих зразків.

5. Big Data [6] – це технологія, що використовується для обробки, збереження та аналізу великих обсягів даних, які зазвичай перевищують можливості традиційних баз даних. Вона включає в себе, інструменти та методи, що дозволяють збирати, зберігати, аналізувати та інтерпретувати великі обсяги структурованих та неструктурованих даних з різних джерел, таких як: сенсорні системи, транзакційні бази даних, соціальні мережі тощо.

Основні характеристики Big Data – це 3V:

– обсяг (Volume) – велика кількість даних, що зберігаються та обробляються;

– різноманітність (Variety) – різноманітність джерел та форматів даних, що збираються;

– швидкість (Velocity) – висока швидкість збирання та обробки даних.

Big Data використовується в різних галузях, включаючи бізнес, медицину, науку, фінанси, громадську безпеку та багато інших. Використання цих технологій, дозволяє отримати значну кількість цінної інформації, яка може допомогти в прийнятті рішень, виявленні тенденцій та передбаченні майбутніх подій [7]. Також, дозволяє знизити ризик помилок та покращити продуктивність діяльності.

6. Інтернет речей (IoT). Дана технологія, дозволяє об'єднати різноманітні фізичні об'єкти та пристрої в єдину мережу, що може збирати, обробляти та передавати дані. Ці об'єкти можуть бути різних типів – від побутової техніки та датчиків смартфонів до великих промислових обладнань.

IoT дозволяє взаємодіяти з фізичними об'єктами та збирати з них дані, що дозволяє отримувати більш повну та точну інформацію про навколишнє

середовище та стан обладнання. Ці дані можуть використовуватися для оптимізації процесів, покращення ефективності роботи обладнання та зниження витрат. Базові принципи «інтернет речей» зображений на рисунку 1.1.



Рисунок 1.1 – Базові принципи Інтернет речей [1]

Технологія IoT також дозволяє автоматизувати ряд процесів та створити нові сервіси та продукти. Наприклад, можна створити систему "розумний дім", в якому будь-який пристрій може бути з'єднаним з мережею та керуватися з смартфона. Вона ще використовується в промисловості для моніторингу та керування виробничими процесами [8].

7. Штучний інтелект. Застосування штучного інтелекту дозволяє модним брендам покращувати процеси продажу та обслуговування клієнтів [9], а також забезпечує ефективнішу роботу з великими об'ємами даних та аналізу модних тенденцій. Класифікація наведена на рисунку 1.2.

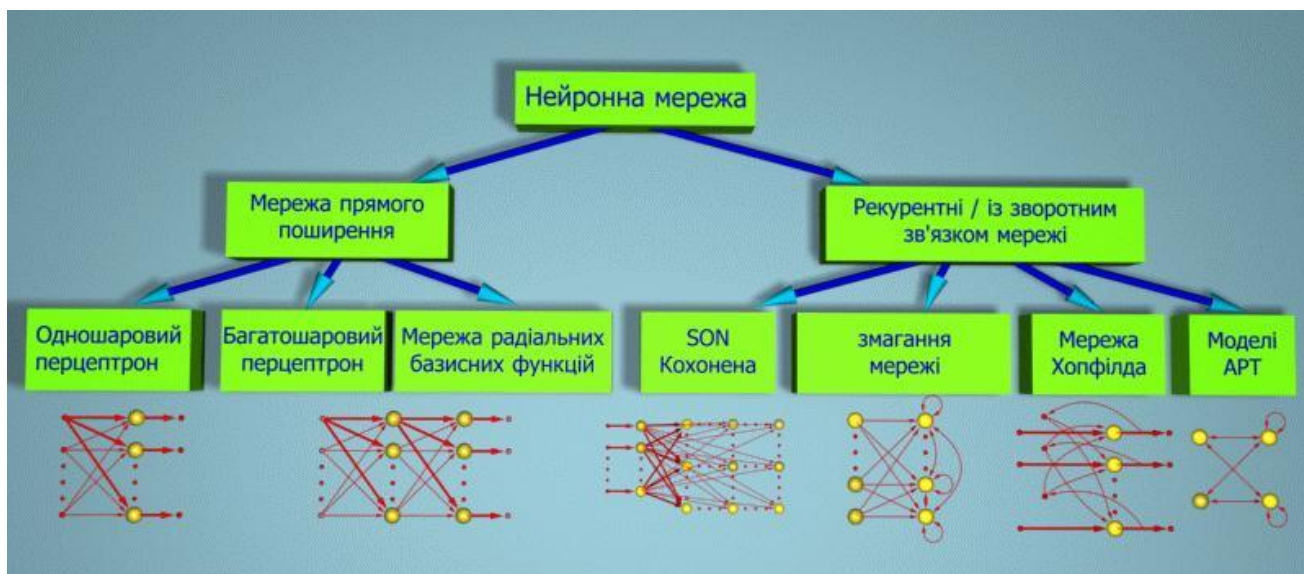


Рисунок 1.2 – Класифікація нейронної мережі [10]

Загалом, застосування інформаційних технологій дозволяє модним брендам підвищити ефективність та продуктивність роботи, покращити взаємодію зі своїми клієнтами та забезпечити високу якість виробів та послуг.

1.2 Аналіз методів, засобів та технологій оброблення та генерування цифрових зображень

Цифрові зображення є результатом захоплення та обробки інформації у формі світла. Створюється за допомогою комп'ютера, сканера або камери. Вони створюються електронними пристроями, такими як камери та сканери, і передається на комп'ютер через кабель або бездротове з'єднання. Їх можна легко зберігати та обробляти. Також можна редагувати та розфарбовувати.

Цифрові зображення використовуються в багатьох галузях, таких як реклама, дизайн, медицина, наука та інші. Для їх оброблення та генерування використовуються різні методи, засоби та технології. Розглянемо деякі з них.

Растрові та векторні графічні редактори. Растрові графічні редактори, такі як Adobe Photoshop, використовують для редагування цифрових зображень, що містять пікселі. Вони дозволяють робити такі операції, як зміна розміру, налаштування кольору та яскравості, видалення певних об'єктів тощо. Векторні

графічні редактори, наприклад Adobe Illustrator, дозволяють створювати векторні зображення, що складаються з математичних об'єктів, таких як лінії, криві та полігони [11]. Методи зображені на рисунку 1.3.



Рисунок 1.3 – Методи оброблення цифрового зображення [10]

3D-модельювання та візуалізація. 3D-модельювання використовують для створення 3D-моделей об'єктів, що використовують в різних галузях, таких як дизайн, архітектура, ігри та інші. Для візуалізації 3D-моделей користуються такими програмами, як: Autodesk 3ds Max, Blender, Cinema 4D та інші.

Масштабування зображень. Ця технологія дозволяє збільшувати розмір зображення без втрати якості. Вона може бути корисною для індустрії моди, де якість самого зображення є важливим, і ці самі зображення можуть бути масштабовані для різних розмірів та пристроїв. Опис технологій та методів описаний в таблиці 1.1.

Таблиця 1.1 – Опис методів і технологій зображень

Технологія/метод	Короткий опис	Переваги	Недоліки
OpenCV [12]	Це бібліотека відкритого коду для обробки зображень та комп'ютерного зору. Вона розроблена для розв'язання задач, пов'язаних з зображеннями, відео, розпізнаванням обличч, знаходженням об'єктів, вимірюванням відстаней та іншими.	1. OpenCV має велику кількість готових функцій для розв'язання задач, обробки зображень та комп'ютерного зору. 2. Ця технологія підтримує багато мов програмування, таких як C++, Python, Java та інші.	1. Для коректної роботи з OpenCV необхідно мати знання з програмування та обробки зображень. 2. Деякі функції OpenCV можуть бути важкими для розуміння та використання для початківців.
CNN/U-net [13]	Convolutional Neural Network (CNN) – це архітектура нейронної мережі, яка використовується для обробки зображень та відео. U-Net – це модифікована версія CNN, яка використовується для задач сегментації зображень.	1. Добре працюють для завдань обробки зображень. 2. Можуть бути навчені на великих наборах даних з високою точністю. 3. Дозволяють автоматично визначати ознаки на зображеннях.	1. Потребують багато ресурсів для навчання. 2. При недостатній кількості даних вони можуть перенавчитися або до навчитися. 3. Не підходять для всіх видів завдань машинного навчання.
GAN [14]	Це архітектура нейронної мережі, яка здатна генерувати нові зображення, звуки. Архітектура GAN складається з двох нейронних мереж:	1. Здатність генерувати нові зображення, звуки та інші дані. 2. Застосування GAN в різних галузях, таких як комп'ютерне зору,	1. Складність побудови та тренування GAN 2. Важкість контролю якості генерованих зображень

	генератора та дискримінатора.	обробка природних мов та синтез голосу.	3. Потреба в великій кількості даних для навчання
--	-------------------------------	---	---

GAN дозволяє генерувати нові зображення та дані з великою точністю, метод OpenCV дозволяє виявляти об'єкти на зображеннях та виконувати різні завдання комп'ютерного зору, а CNN/U-Net – це архітектури нейронних мереж [15], що використовуються для обробки зображень та відео.

У підсумку, індустрія моди активно використовує цифрові зображення та різноманітні технології обробки та генерування. Їх використання, дозволяє створювати нові продукти та послуги, які задовольняють потреби споживачів та покращують ефективність бізнесу.

1.3 Використання нейронних мереж для віртуальної примірки одягу

Примірка є важливою частиною дизайну одягу. Це процес, який допомагає вам знайти правильний розмір для одягу, генеруючи серію математичних векторів. Програмне забезпечення для віртуального підбору – чудовий спосіб для дизайнерів випробувати свої концепції, не витрачаючи фізичних ресурсів. Однак більшість методів підгонки все ще виконуються вручну. Це може призвести до неточних результатів і затримок у циклах розробки.

Генеративні нейронні мережі (ГНМ) [14] – це новий спосіб для створення відповідних моделей. Дані системи працюють, навчаючи кілька мереж, які використовують дані, з генеровані з реального одягу. Кожен вузол у мережі представляє елемент дизайну одягу, і його можна навчити розраховувати вектор для конкретної форми тіла. Потім цей вектор застосовується до вихідного одягу для вимірювання. По суті, це створює індивідуальний одяг безпосередньо з математичних моделей.

Генеративні нейронні мережі, можуть бути використані в індустрії моди для створення віртуальних примірок одягу. ГНМ – це глибокі нейронні мережі

[16], які можуть навчатися генерувати нові зображення, що дуже схожі на зображенні з навчальної вибірки, вона показана на рисунку 1.4.

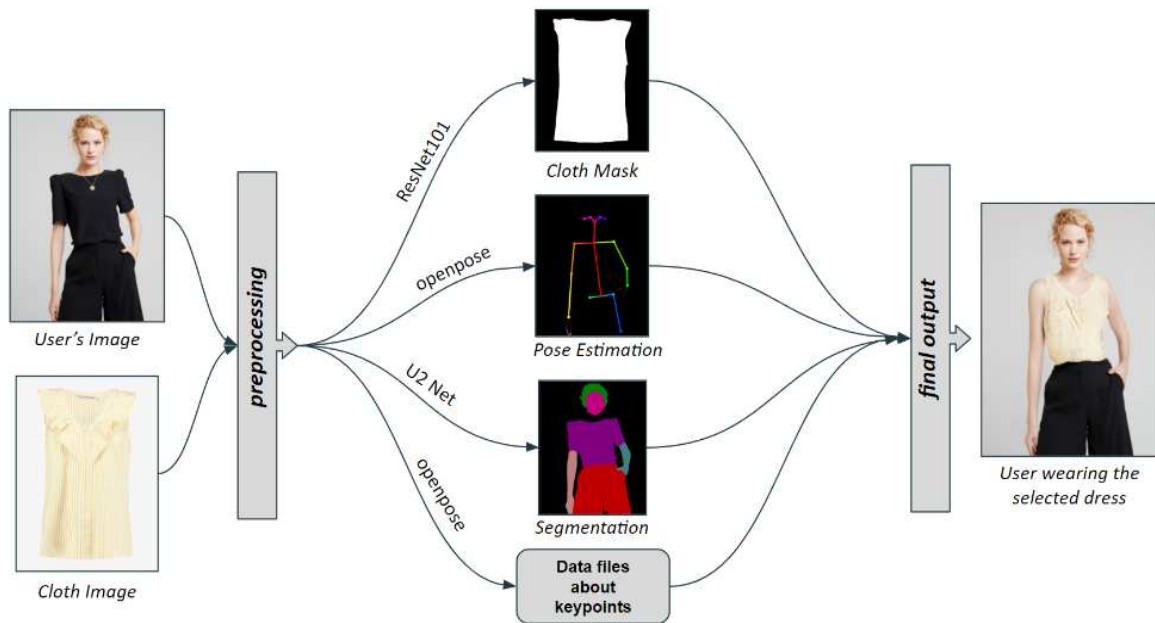


Рисунок 1.4 – Примірка одягу з використанням ГНМ [17]

З їх використанням можна створювати віртуальні примірки одягу, які дуже реалістично відображають вигляд і фітинг одягу на людині. Навчання ГНМ відбувається з використанням великої кількості зображень людей у різному одязі та з різними фізичними параметрами, що дозволяє моделі навчатися розпізнавати різні типи одягу та призначати його розмір та вигляд на основі вхідних даних про користувача.

Для створення віртуальної примірки одягу за допомогою ГНМ можуть бути використані різні архітектури мереж, наприклад, Deep Convolutional GAN (DCGAN), Conditional GAN (CGAN) та інші. У DCGAN використовується глибокий згортковий шар для визначення функцій активації, що дозволяє створювати більш складні зображення, такі як фотографії людей у різному одязі. У CGAN використовуються умовні вхідні дані, що дозволяє створювати віртуальні примірки одягу, які більш точно відповідають фізичним параметрам користувача [17]. Опис різних методів GAN розписаний у таблиці 1.2.

Таблиця 1.2 – Показ та опис різних методів GAN

Метод GAN	Короткий опис	Переваги	Недоліки
DCGAN [18]	DCGAN (Deep Convolutional Generative Adversarial Network) – це модифікована версія GAN, яка використовує в своїй архітектурі згорткові шари.	<p>1.Добре працює для завдань генерації зображень.</p> <p>2.Може генерувати високоякісні зображення з високою роздільною здатністю.</p> <p>3.Дозволяє створювати реалістичні зображення на основі навчальних даних.</p>	<p>1.Вимагає великої кількості навчальних даних та обчислювальних ресурсів.</p> <p>2.Може бути складно налаштувати та навчити.</p> <p>3.При недостатній кількості даних може перенавчитися або до навчитися.</p>
CGAN [19]	CGAN (Conditional Generative Adversarial Network) – це розширення GAN, яке дозволяє генерувати зображення з урахуванням певних умов.	<p>1.Дозволяє генерувати зображення з урахуванням додаткової інформації, наприклад, кольору, форми або розміру.</p> <p>2.Може бути використаний для генерації зображень з вищою якістю та з урахуванням конкретних потреб користувача.</p>	<p>1.Вимагає більш складної архітектури та навчання, ніж звичайний GAN.</p> <p>2.Потребує більше обчислювальних ресурсів та часу для навчання.</p> <p>3.Під час генерації зображень з використанням CGAN можуть виникати проблеми з перенавчанням, якщо вхідні умови не є репрезентативними для всіх можливих варіантів.</p>
StyleGAN [20]	Це генеративно-змагальна мережа, яка використовує техніку	Здатність генерувати зображення з високою якістю, зберігаючи	1.Складність процесу навчання та вимоги до потужності

	<p>"передачі стилю", щоб генерувати зображення з високою якістю та різноманітністю стилів.</p>	<p>деталі та текстуру, а також різноманітності згенерованих зображень.</p>	<p>обчислювального обладнання. 2. Моделі можуть страждати від проблеми "сильного" передачі стилю, яка може призводити до зміни змісту зображення.</p>
--	--	--	---

Для навчання ГНМ можуть бути використані різні джерела даних, наприклад, фотографії моделей, 3D-моделі одягу та інші. Зазвичай, навчальні дані містять велику кількість зображень, що дозволяє їм навчатися на різноманітних типах одягу та фізичних параметрах користувачів.

Проте, використання ГНМ для віртуальної примірки одягу має свої обмеження. Наприклад, створені віртуальні примірки можуть не завжди точно відображати фізичну структуру одягу, таку як дрібні деталі та текстиль, що може призвести до неточності у примірці [14, 17, 21]. Робота загальної віртуальної примірки показано на рисунку 1.5.

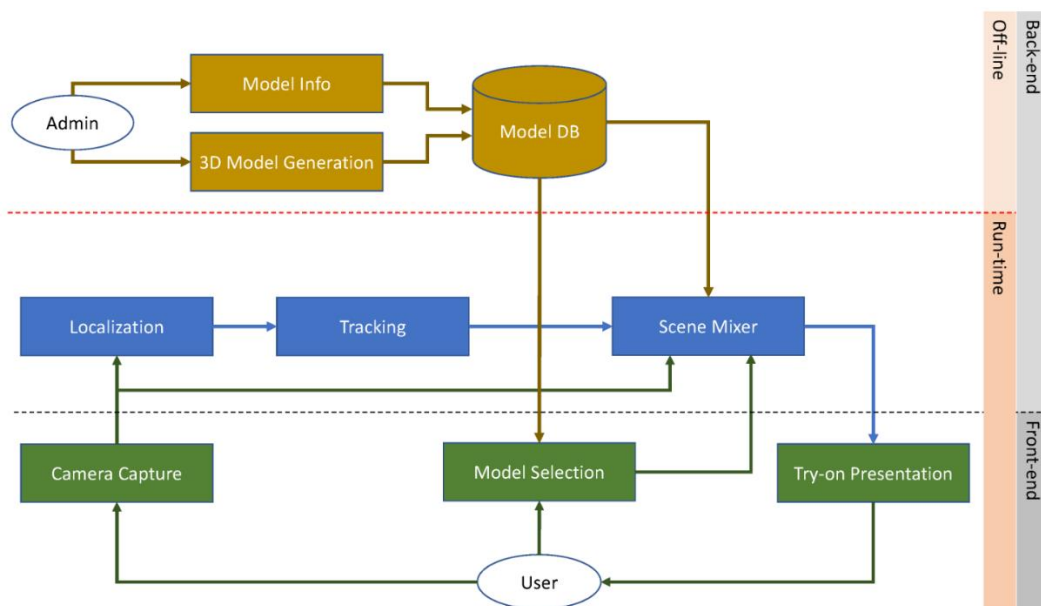


Рисунок 1.5 – Робота загальної віртуальної системної примірки [21]

Також, існують певні технічні труднощі. Наприклад, зображення, отримані з ГНМ, можуть мати низьку якість або бути спотвореними, що може ускладнити процес віртуальної примірки [22]. Крім того, ГНМ можуть бути чутливі до змін в світлі та тіней, що теж може бути причиною неточної примірки.

Загалом, використання генеративних нейронних мереж для віртуальної примірки одягу є перспективним напрямком у розвитку модної індустрії. Віртуальна примірка, дозволяє користувачам більш точно відображати фізичні параметри та вибрати правильний розмір одягу, що зменшує ймовірність повернення товару та ризик незадоволення покупкою.

1.4 Мета, завдання та вимоги до реалізації інформаційної системи

У галузі електронної комерції основною перешкодою, з якою стикаються споживачі, є неможливість приміряти одяг перед покупкою, що призводить до високого рівня повернень, незадоволення клієнтів і значних фінансових втрат для бізнесу. Наявні рішення для віртуальної примірки є обмеженими, неточно відображають посадку, стиль і зовнішній вигляд людини на зображенні з її унікальним типом фігури. Такий користувацький досвід призводить до браку впевненості та небажання споживачів здійснювати покупки одягу онлайн. Тому задача, що розглядається, полягає в покращенні візуального сприйняття та реалістичності цифрової примірки одягу, включно з реалістичним відображенням тканини одягу, посадки, драпірування і взаємодії з позами людського тіла, через реалізацію інформаційної системи на основі методів та засобів штучного інтелекту, що дасть змогу виконувати якісну віртуальну примірку одягу.

Отже, метою кваліфікаційної роботи бакалавра є покращення віртуальної примірки одягу за цифровими зображеннями людини та предметів одягу. Для досягнення поставленої мети необхідно виконати наступні завдання:

1. Провести аналіз засобів штучного інтелекту для оброблення та генерування цифрових зображень та обрати найкращий.

2. Використати обраний засіб оброблення та генерування цифрових зображень для розв'язання задачі віртуальної примірки за цифровими зображеннями людини та предметів одягу.

3. Реалізувати спосіб віртуальної примірки одягу у вигляді модуля програмного забезпечення.

4. Провести експериментальне тестування реалізованого модуля за еталонними наборами даних.

Оцінювання створеного способу та реалізованої інформаційної системи на його основі має відбуватися через візуальне сприйняття накладеного одягу на зображення людини в різних позах та наявність візуальних артефактів на новостворених зображеннях.

Розділ 2 Спосіб віртуальної примірки одягу з використанням генеративних нейронних мереж

2.1 Спосіб формалізації та оброблення цифрового зображення

Крок збору даних про одяг є одним з перших та найважливіших етапів при створенні його віртуальної примірки. Його метою є забезпечення достатньої кількості даних, які необхідні для подальшої обробки та аналізу. Основні етапи збору даних про одяг включають наступні дії:

1. Визначення потрібної кількості даних. Перш за все, необхідно визначити, яка кількість даних буде достатньою для подальшої обробки та аналізу. Це може залежати від конкретної мети та задачі, для яких буде використовуватися віртуальна примірка.

2. Зйомка фото або відео. Для збору даних про одяг необхідно зробити фото або відео з різних ракурсів та в різних варіантах. Це може включати його зйомку на моделі або на манекені.

3. Обробка даних. Після зйомки даних необхідно їх обробити для подальшого використання. Це може включати ретушування та підготовку фотографій, видалення фону, корекцію кольору та інші дії.

4. Анотування даних. Після обробки даних необхідно їх анотувати, тобто позначити різні характеристики одягу на зображенні. Наприклад, розмір, фасон, матеріал, колір тощо. Це допомагає забезпечити точність та якість подальшої обробки.

5. Збереження даних. Після анотування необхідно їх зберегти у відповідному форматі для подальшого використання. Зазвичай це можуть бути текстові файли з описом характеристик одягу та посилання на відповідні фото чи відео.

6. Оцінка якості даних. Перед використанням зібраної інформації необхідно оцінити її якість та достовірність. Для цього можуть бути використані різні методи та алгоритми, такі як візуальна оцінка зображень, перевірка на наявність помилок у вказаних характеристиках та інші.

7. Використання даних. Після збору, обробки, анотування та оцінки якості, вони можуть бути використані для створення віртуальної примірки одягу. Це може включати розробку алгоритмів для відтворення форми, фасону та розміру одягу на віртуальній моделі, а також створення програмного забезпечення для візуалізації та тестування одягу на віртуальній моделі.

Після збору, обробки, анотування та оцінки якості даних їх можна використовувати для створення віртуальної примірки одягу. Це може бути важливим етапом у моделюванні та візуалізації модного одягу, а також для виробництва одягу, де віртуальна примірка може служити для визначення форми, фасону, розміру та інших аспектів одягу.

Для відтворення форми, фасону та розміру одягу на віртуальній моделі можуть бути розроблені алгоритми та методи, які використовують інформацію, що була зібрана та анотована раніше. Ці алгоритми можуть брати до уваги різні параметри, такі як розміри тіла, анатомічні особливості, тканини одягу та інші фактори, щоб точно відтворити вигляд та паспортні дані одягу на віртуальній моделі.

Для візуалізації та тестування одягу на віртуальній моделі може бути розроблене спеціальне програмне забезпечення. Це дозволить користувачам взаємодіяти з віртуальною приміркою одягу, спробувати різні комбінації, перевірити вигляд та функціональність, тестувати вплив різних параметрів на одяг. Таке програмне забезпечення може допомогти вдосконалити процес проектування, підвищити точність та зручність фітінгу одягу, а також покращити взаємодію з клієнтами.

Використання даних для створення віртуальної примірки одягу може сприяти прискоренню процесу розробки та виробництва одягу, зменшенню витрат та ефективному управлінню асортиментом. Віртуальна примірка дозволяє візуалізувати та оцінити різні дизайни та варіації одягу перед фізичним виготовленням, що забезпечує більш точні та ефективні рішення в галузі моди та текстилю.

Узагальнюючи, збір даних про одяг є важливим етапом при створенні віртуальної примірки одягу. Це включає зйомку фото або відео одягу, обробку та анування даних, оцінку якості даних та їх використання для створення віртуальної примірки.

Форматування зображення – це процес обробки зображення з метою зміни його розміру та формату, щоб його можна було легко використовувати для віртуальної примірки або для інших цілей [16].

Основні кроки візуального відтворення:

– Вибір формату: залежно від призначення (віртуальна примірка, веб-сторінка, друк тощо) вибирається відповідний формат зображення. Найпоширенішими форматами є JPEG, PNG, GIF, BMP та TIFF.

– Зміна розміру: зображення може бути зменшене або збільшене до необхідного розміру. При зменшенні розміру може втратити деталі, тому необхідно використовувати алгоритми, які дозволять зберегти максимальну якість. При збільшенні можуть бути додані нові пікселі, що може привести до розмиття та зниження якості картинки, тому слід використовувати алгоритми збільшення з використанням інтерполяції.

– Обрізка: якщо зображення має надмірну площу або непотрібну частину, його можна обрізати, щоб зберегти лише потрібну частину.

– Конвертація кольорів: залежно від формату графічного зображення та призначення, може знадобитись конвертування. Наприклад, для веб-сторінки може бути необхідним конвертувати зображення в формат RGB.

Додавання ефектів: зображенню можуть бути додані різні ефекти, такі як зменшення шуму, насиченість кольорів, розмиття, фільтри та інші ефекти, що дозволяють поліпшити якість картинки.

Компресія: після форматування зображення, можна зменшити його розмір та зберегти місце на диску. Для цього потрібно використовувати різні алгоритми стиснення, такі як безвтратне або ж стиснення з втратами.

Збереження: після завершення форматування зображення його необхідно зберегти на диску у відповідному форматі. Зазвичай, для цього використовують

формат JPEG для фотографій, PNG для зображень з прозорим фоном та GIF для анімацій.

У процесі форматування зображення важливо дотримуватись правильної послідовності кроків, щоб зберегти якість картинки та отримати необхідний формат та розмір. Також, слід пам'ятати про те, що при збільшенні розміру може відбутись втрата якості, тому краще використовувати великі візуальні відтворення та зменшувати їх розмір до необхідних параметрів.

Розпізнавання контуру одягу – це процес визначення границь одягу на зображенні. Цей процес можна розбити на кілька етапів:

1. Попередній аналіз: перед початком розпізнавання контуру одягу необхідно виконати цю дію. Це може включати в себе зменшення шуму та регулювання яскравості та контрастності зображення.

2. Виявлення границь: для виявлення границь одягу можна використовувати алгоритми фільтрації, такі як фільтр «Собеля» або фільтр «Канні». Ці алгоритми визначають градієнти зображення та виявляють контури об'єктів на зображенні.

3. Бінаризація – це процес перетворення зображення з відтінками сірого у чорно-біле зображення, де кожен піксель може бути або чорним, або білим. Бінаризація зображення, зазвичай використовується для покращення якості картинки та полегшення подальшої обробки. Бінаризація зображена на рисунку 2.1.



Рисунок 2.1 – Результати впровадження бінаризації до зображення [12]

Порогова фільтрація – це один з методів бінаризації, який базується на завданні певного порогового значення, яке розділяє пікселі на дві групи: пікселі з

яскравістю вище порогового значення стають білими, а пікселі з яскравістю нижче порогового значення – чорними.

Порогова фільтрація є простим, але ефективним методом обробки зображень, який використовується для бінаризації зображень. Задача порогової фільтрації полягає у визначенні певного порогового значення, яке відокремлює пікселі зображення на дві категорії: пікселі, що мають яскравість вище порогового значення, стають білими, тоді як пікселі з яскравістю нижче порогового значення стають чорними.

Процес порогової фільтрації може бути застосований до зображень з різними типами даних, такими як відео, фотографії або зображення, отримані з сенсорів. Він використовується для розділення областей зображення на основі інтенсивності пікселів і забезпечує простий спосіб отримання двійкових зображень, що використовуються в подальшому аналізі та обробці зображень.

Вибір правильного порогового значення є важливим кроком у пороговій фільтрації. Якщо порогове значення вибране неправильно, можуть виникнути проблеми, такі як втрата деталей або неправильне визначення об'єктів на зображенні. Тому необхідно проводити аналіз зображень та експериментувати з різними пороговими значеннями для досягнення оптимальних результатів.

Незважаючи на свою простоту, порогова фільтрація залишається потужним інструментом для вирішення багатьох задач обробки зображень, включаючи сегментацію об'єктів, виявлення контурів, розрізнення фону та об'єктів та багато інших. Вона широко використовується в різних галузях, таких як медицина, комп'ютерний зір, контроль якості та багато інших застосувань.

Для застосування порогової фільтрації необхідно спочатку вибрати певне значення. Це може бути, наприклад, середнє значення яскравості пікселів на зображенні, або ж задане користувачем. Після цього кожен піксель на картинці порівнюється з пороговим значенням. Якщо яскравість пікселя більше, то він стає білим, якщо менше – чорним.

Отже, бінаризація – це важливий етап в обробці зображень, який дозволяє виділити потрібні деталі та покращити якість для подальшої обробки.

1. Видалення шуму. В обробці зображень, шум – це небажані пікселі на картинці, які можуть заважати виявленню різних об’єктів та викликати неточності у подальшій обробці. В контексті виявлення границь одягу, на бінарному зображенні можуть з’являтися невеликі області білого або чорного кольору, які не належать до об’єкту (одягу) і їх потрібно видалити. Видалення шуму показано на рисунку 2.2.

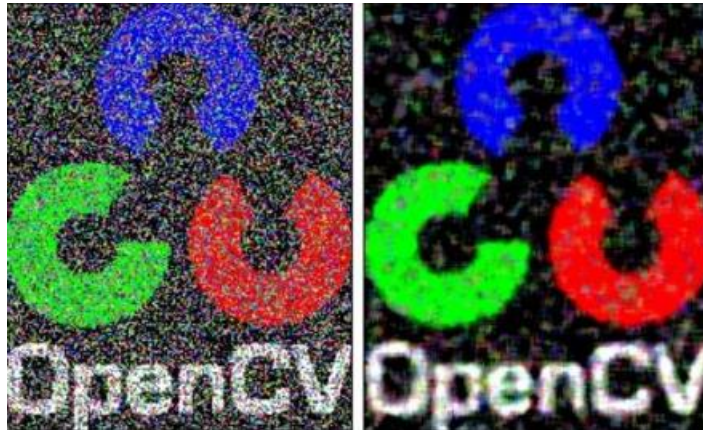


Рисунок 2.2 – Видалення шуму з використанням OpenCV [12]

Для цього потрібно на бінарному зображенні використати алгоритм морфологічної обробки, який базується на математичних операціях зміни форми та структури об’єктів на зображенні. Два основних типи операцій морфологічної обробки – це розширення та звуження.

Операція розширення, має на меті збільшити об’єкти на зображенні шляхом додавання пікселів по контуру об’єкта. Це дозволяє заповнити дрібні отвори або з’єднати окремі частини об’єкту, що допомагає зменшити розмір шуму та збільшити розмір об’єкту.

Операція звуження, має на меті зменшити різні предмети на зображенні шляхом видалення пікселів по контуру об’єкта. Це дозволяє видалити дрібні виступи або зробити об’єкт більш компактним – це допоможе видалити шум та зменшити розмір.

При використанні морфологічних операцій необхідно враховувати, що вони можуть змінювати форму та розмір об'єкта. Тому важливо виконувати їх з розумінням та тестувати на ефективність з різними типами зображень.

2. Виявлення контуру. Після видалення шуму, можна виявити контур одягу. Для цього можна використовувати алгоритми виявлення контуру, такі як алгоритм «Сью», який визначає зовнішні та внутрішні контури об'єкту на візуальному відтворенні.

3. Визначення форми: після виявлення контуру, можна визначити форму вбрання. Це може бути зроблено за допомогою алгоритмів аналізу, таких як алгоритм «Рамер-Дугласа-Пекера», який дозволяє зменшити кількість точок, які визначають контур об'єкту, не втрачаючи при цьому його форми.

4. Сегментація. Після виявлення контуру одягу на зображенні та видалення шуму можна перейти до процесу сегментації, який допоможе розділити зображення на окремі частини відповідно до їхнього значення і призначення. Цей процес може бути виконаний за допомогою різних методів, таких як порогова сегментація, регіональна сегментація, та сегментація за допомогою графів.

Порогова сегментація – це метод, який використовує пороговий підхід до розділення зображення на окремі частини. Цей метод полягає в тому, що на картинці вибирається піксель або група пікселів, які будуть використовуватися як поріг для визначення кольору кожної частини. Іншими словами, порогова сегментація використовує порогові значення для розділення на області зі схожими властивостями, такими як кольори.

Регіональна сегментація – це метод, який використовується для розділення зображень на регіони зі схожими властивостями. Цей метод полягає в тому, що спочатку на картинці вибирається піксель, що належить до певного класу, і потім аналізується його сусідство для визначення регіону з ідентичними властивостями.

Сегментація за допомогою графів – це метод, який використовується для визначення регіонів на зображенні, використовуючи побудову графів. Цей метод полягає в тому, що на зображенні будуються графи, де вершини відповідають пікселям, а ребра відображають схожість між ними.

Крім цього, можна використовувати алгоритми кластеризації, такі як k-means, для розділення пікселів зображення на групи в залежності від їхньої яскравості або кольору. Таким чином, їх можна розділити на окремі регіони, які відповідають різним елементам одягу.

Отже, сегментація є важливим етапом обробки, оскільки дозволяє виділити окремі частини та працювати з ними окремо. В результаті можна отримати детальнішу інформацію про форму, розташування та взаємне розміщення різних елементів одягу на зображенні.

В цілому, розпізнавання контуру одягу – це важливий етап в процесі створення віртуальної примірки. Він дозволяє точно визначити форму вбрання на зображенні та підготувати візуальне відтворення для подальшої обробки та аналізу.

Візуалізація – це процес створення візуального відображення одягу на екрані комп'ютера або іншого пристрою. Цей процес може бути виконаний за допомогою різноманітного програмного забезпечення, такого як: 3D-моделювання, програма для рендерингу, програми для віртуальної примірки та інші.

Перш за все, модель вбрання повинна бути імпортована в програму для візуалізації, яка дозволить відображати його в режимі реального часу. Далі віртуальна примірка дозволить взаємодіяти з одягом, змінюючи його розмір, кольори, форму та інші параметри.

Користувач може вибрати різні варіанти моделі, додавати та видаляти деталі, змінювати кольори та тканини, а також виконувати інші дії, що дозволяють налаштовувати модель на свій смак.

У програмах для віртуальної примірки одягу можуть бути встановлені різні функції, такі як анімація, що дозволяє переглядати модель з різних кутів з різних точок зору, або можливість зберігати вигляд вбрання в зображенні, що дозволить поділитися знімками своєї віртуальної примірки з іншими користувачами.

У результаті, віртуальна примірка дозволяє побачити, як буде виглядати одяг на людині, змінювати його, попередньо переглядаючи варіанти, і відправляти модель на виробництво або друк на 3D-принтері.

2.2 Проєктування архітектури генеративної нейронної мережі для віртуальної примірки одягу

Проєктування архітектури генеративної нейронної мережі для віртуальної примірки одягу можна розбити на такі етапи:

1. Збір даних. Перший етап полягає тому, що буде збиратися велика кількість фотографій одягу з різним освітленням та ракурсів. Для кожної фотографії потрібно зберегти додаткову інформацію про розмір, кольори та інші параметри. Також потрібно зберегти фотографії людей, які носять цей одяг, щоб мати можливість простежити, як він виглядає на різних фігурах.

2. Попередня обробка даних. Після першого етапу (збору даних) потрібно провести попередню обробку. Це може включати видалення шуму з фотографій, вирівнювання розмірів та форматування даних у зручний для подальшої обробки формат.

3. Архітектура DCGAN. Вона має дві складові мережі – генератор та дискримінатор.

Генератор приймає на вхід вектор з шумовими даними і трансформує їх в зображення, що максимально наближене до зображень з навчального набору даних.

Дискримінатор приймає на вхід зображення та визначає ймовірність того, що воно належить до навчального набору даних.

У процесі навчання ГНМ генератор та дискримінатор взаємодіють між собою у формі двобічної гри. Генератор намагається покращити свої навички, щоб згенерувати більш реалістичні зразки, які дискримінатор буде важко розрізнити від реальних. У той же час, дискримінатор намагається навчитися відрізнити між синтезованими зразками і реальними зразками з якомога більшою точністю.

Процес навчання ГНМ включає послідовні ітерації, під час яких генератор і дискримінатор навчаються і покращують свої навички. Генератор отримує зворотний зв'язок від дискримінатора щодо якості його синтезованих зразків і використовує цю інформацію для покращення результатів. Дискримінатор, у свою чергу, навчається розрізняти між реальними і синтезованими зразками, а також адаптується до змін у якості генерованих зразків.

Процес навчання ГНМ триває досягнення балансу між генератором і дискримінатором, коли генеровані зразки стають максимально реалістичними, а дискримінатор не може ефективно відрізнити їх від реальних. У такому випадку генератор досягає своєї мети створення реалістичних зразків.

У роботі спроектовано підхід на основі ГНМ, який дає змогу на основі вхідних даних одягу і людини генерувати нове цифрове зображення (рисунок 2.3).

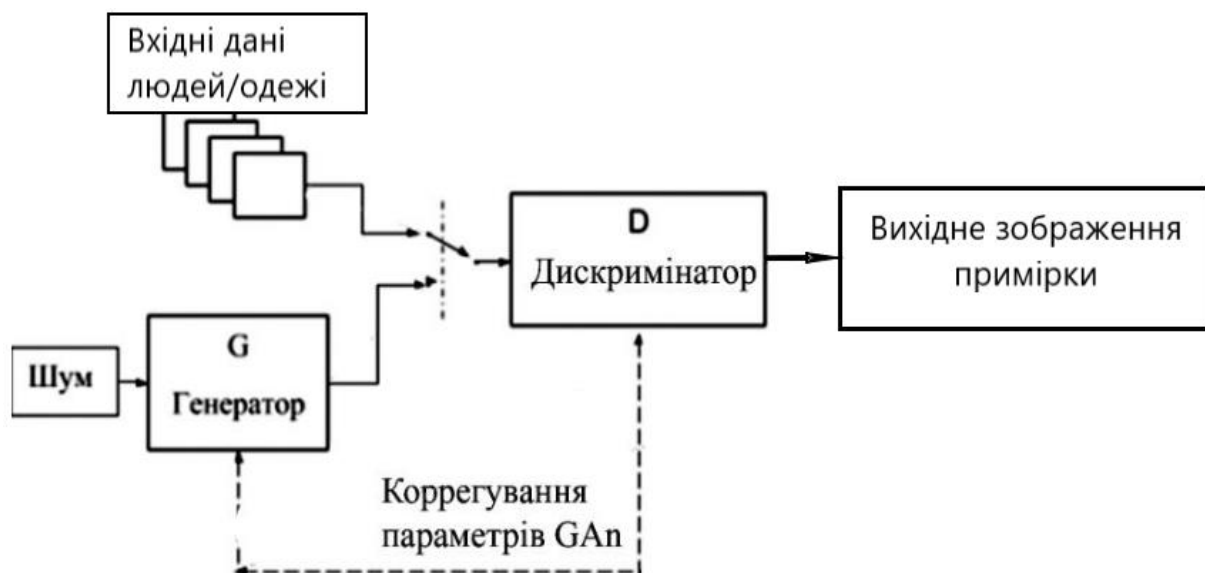


Рисунок 2.3 – Архітектура ГНМ

Водночас процес навчання ГНМ за підходом з рисунку 2.4 може бути складним і вимагати великої кількості тренувальних даних, налаштування гіперпараметрів та використання певних методів оптимізації для досягнення стабільного навчання ГНМ. Однак, коли ГНМ успішно навчається, вона може бути використана для генерації нових, реалістичних зразків, що має широкі

застосування в області комп'ютерного зору, графіки та інших сферах.. Показ архітектури DCGAN наведено у рисунку 2.4.

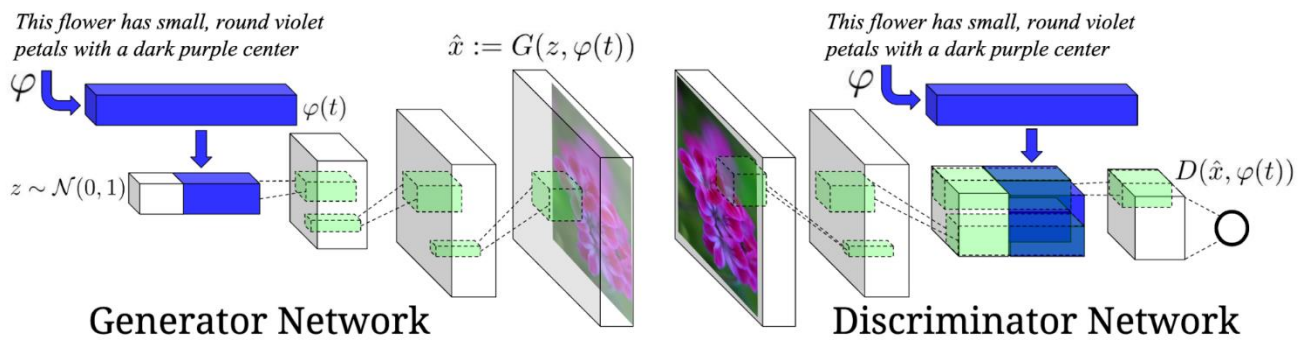


Рисунок 2.4 – Використана у роботі архітектура ГНМ

1. Тренування мережі. Після розробки архітектури мережі потрібно провести її тренування на зібраних даних. В цьому процесі мережа буде навчатися генерувати віртуальні примірки, які максимально схожі на реальні зразки. Тренування мережі може займати декілька годин або навіть днів, залежно від складності архітектури та кількості даних.

2. Оцінка результатів та покращення мережі. Після тренування необхідно оцінити результати та виявити можливі недоліки. Можливо, що мережа не генерує достатньо деталізовані зразки одягу або недостатньо точно відтворює кольори та фактуру тканини. У такому випадку потрібно провести покращення мережі, змінити архітектуру та додати додаткові дані для тренування.

3. Використання мережі. Після успішного тренування та покращення мережі можна починати використовувати її для генерації віртуальної примірки вбрання. Користувачі можуть вибрати тип одягу, розмір, колір, а мережа згенерує його віртуально. За бажанням, можна буде переглянути в 3D-режимі та оцінити, як буде виглядати реальний одяг на конкретній фігурі. Є можливість приміряти віртуальний одяг за допомогою програми доповненої реальності.

Після успішного навчання та вдосконалення мережі користувачі можуть почати використовувати її для створення віртуального одягу. Є можливість вибрати тип одягу, розмір, колір і мережа згенерує віртуальну копію. При бажанні

його можна переглянути в режимі 3D, щоб оцінити, як буде виглядати справжній одяг.

Проектування архітектури мережі для складається зі збору даних, попередньої обробки, розробки, тренування, оцінки результатів покращення та використання її для генерації віртуальної примірки. Важливо також зазначити, що успіх проектування мережі залежить від якості та кількості зібраних даних, тому важливо забезпечити максимальну кількість зібраних даних з різних джерел.

У процесі проектування архітектури мережі, також важливо враховувати потреби користувачів та додаткові функції, які можуть збільшити та покращити зручність, користь, та продуктивність віртуальної примірки. Наприклад, можна додати функцію вибору матеріалу, додаткових деталей або різне, гнучке налаштування параметрів фігури для точнішого відображення одягу.

Загалом, проектування цієї архітектури є складним та багатоетапним процесом, який може принести значну користь та зручність багатьом користувачам, які шукають якусь віртуальну альтернативу для реальної примірки одягу.

2.3 Функціональна структура інформаційної системи на основі генеративної нейронної мережі

Інформаційна система на основі ГНМ матиме наступну функціональну структуру:

1. Збір та підготовка даних. Ця функція включає збір та підготовку вхідних даних для нейронної мережі. Дані можуть бути зібрані з різних джерел, наприклад, з баз даних, файлів, сенсорів або Інтернету. Підготовка даних включає попередню обробку, таку як нормалізацію, шкалювання, очищення та інші маніпуляції з даними, які необхідні для подальшої роботи з ними.

2. Основний принцип навчання ГНМ полягає у використанні методу зворотного поширення помилок (backpropagation) для оптимізації параметрів

моделі. Зазвичай це включає два основних етапи: пряме поширення (forward propagation) та зворотне поширення помилок.

3. Під час прямого поширення ГНМ виконує обчислення на графі, передаючи інформацію з одного вузла до іншого згідно з вагами та архітектурою мережі. Цей процес можна уявити як поширення сигналу через граф, де кожен вузол обчислює свій внутрішній стан на основі вхідних даних та ваг.

4. Після прямого поширення використовується алгоритм зворотного поширення помилок для розповсюдження помилки від виходу моделі до її входу. Цей процес дозволяє оновити ваги мережі, зменшуючи помилку між прогнозованими та очікуваними значеннями. Зворотне поширення помилок використовує градієнти, обчислені за допомогою функції втрати, для зміни ваг та покращення точності прогнозу.

5. Після завершення процесу навчання ГНМ може використовуватися для генерації нової інформації на основі вхідних даних. Вона може прогнозувати майбутні значення в графі, класифікувати вузли або ребра, виконувати кластеризацію або виконувати інші завдання, залежно від призначення та конфігурації моделі.

6. Важливо відзначити, що для успішного навчання ГНМ необхідні відповідні та якісні навчальні дані, а також правильний вибір архітектури мережі та гіперпараметрів. Також варто враховувати специфічні вимоги та особливості задачі, для якої використовується ГНМ, для досягнення найкращих результатів.

7. Збереження та обробка результатів. Після генерації нових даних, їх необхідно зберегти та обробити для подальшого використання. Ця функція включає збереження результатів у базі даних або у вигляді файлів, а також обробку результатів, яка може включати фільтрацію, вибірку та інші маніпуляції з даними. Показ методу сегментації зображено на рисунку 2.5.



Рисунок 2.5 – Функціональна схема методу сегментації зображень.

8. Взаємодія з користувачем. Інформаційна система на основі ГНМ може включати взаємодію з користувачем, наприклад, через графічний інтерфейс. Користувач може запитувати систему на генерацію нових даних або перегляд результатів обробки даних.

Через графічний інтерфейс (GUI), користувачі можуть здійснювати взаємодію з системою шляхом використання графічних елементів. Наприклад, користувач може завантажувати дані, встановлювати параметри обробки або генерації, та переглядати результати на екрані. GUI забезпечує зручну та інтуїтивно зрозумілу взаємодію з системою для користувачів з різними рівнями технічної підготовки.

За допомогою взаємодії з користувачем, інформаційна система на основі ГНМ може відповідати на запити користувачів, генерувати нові дані, виконувати аналіз або прогнозування, і забезпечувати користувачам доступ до високоякісних результатів обробки даних, заснованих на потужних здатностях ГНМ. Структура інформаційної системи показано на рисунку 2.6.

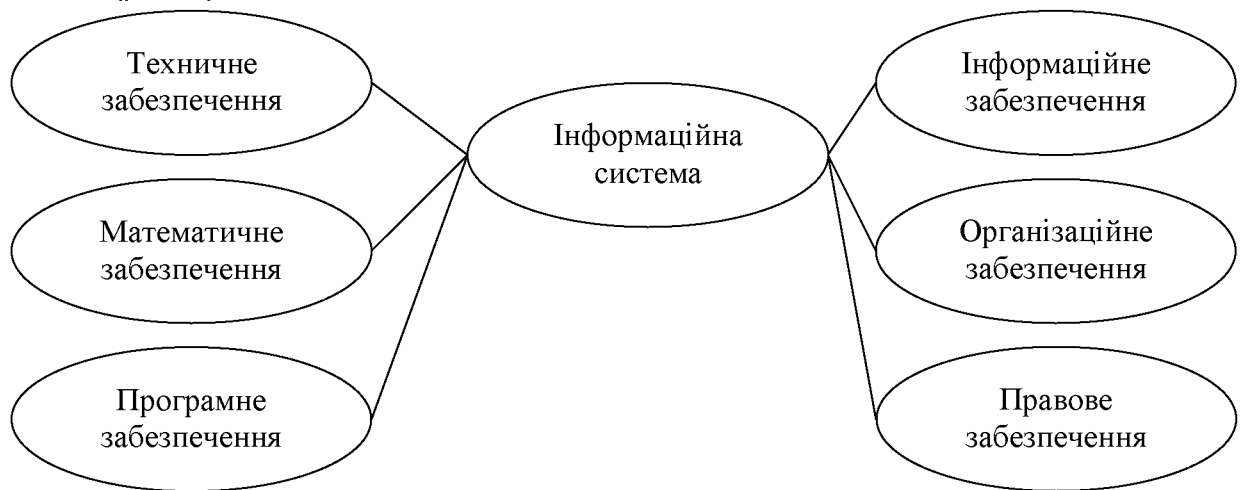


Рисунок 2.6 – Структура інформаційної системи

Взаємодія з користувачем може включати наступні етапи:

- вибір типу одягу для генерації віртуальної примірки;
- налаштування параметрів фігури користувача (розмір, статура тощо);
- перегляд віртуальної примірки одягу з різних ракурсів;
- вибір матеріалу та додаткових деталей одягу.
- збереження результатів та можливість повторно використовувати.

Для забезпечення взаємодії з користувачем можуть використовуватися різні інтерфейси, такі як веб-сайт або мобільний додаток. Важливо забезпечити зручність використання та високу якість віртуальної примірки, щоб користувачі могли зручно та точно оцінити вигляд одягу на собі.

Усі ці вище описані функції можуть бути інтегровані в одну інформаційну систему на основі ГНМ. Вона може використовуватись для різних завдань, таких як генерація музики, зображень або текстів, прогнозування ринкових тенденцій та

інших. Однак, щоб забезпечити її правильну на основі ГНМ, необхідно мати відповідний рівень знань у галузі машинного навчання та нейронних мереж.

2.4 Проектування структури інформаційної системи

2.4.1 Інфологічна модель даних

Проектування структури інформаційної системи починається з розробки інфологічної моделі, яка є концептуальною моделлю даних, що описує сутності, взаємозв'язки та атрибути, які необхідні для функціонування системи.

Вона дозволяє визначити сутності, що зберігаються в інформаційній системі, їхні зв'язки та взаємодії. Це включає у себе визначення таблиць бази даних, поля таблиць та зв'язки між ними.

Після розробки інфологічної моделі даних, можна перейти до фізичної моделі, яка визначає, як інформація буде зберігатися та організовуватися на рівні конкретної бази даних. Фізична модель даних включає в себе структуру таблиць, індекси та інші параметри, необхідні для зберігання та роботи з даними.

Після розробки інфологічної моделі даних, фізична модель визначає, як ці дані будуть зберігатися та організовуватися на рівні конкретної бази даних. Фізична модель даних враховує технічні аспекти збереження та обробки даних, забезпечуючи ефективність та швидкодію системи.

Основні компоненти фізичної моделі даних включають:

- Структура таблиць: Фізична модель визначає, як дані будуть організовані у вигляді таблиць у базі даних. Це включає визначення назв таблиць, стовпців та типів даних для кожного стовпця.

- Індекси: Індекси використовуються для прискорення пошуку та фільтрації даних у базі даних. Фізична модель визначає, які стовпці повинні бути індексовані та які типи індексів будуть використовуватися (наприклад, В-дерева, хеш-таблиці тощо).

- Відношення між таблицями: Якщо є багато таблиць, фізична модель визначає, як вони будуть пов'язані між собою за допомогою ключів і зовнішніх

ключів. Це дозволяє встановлювати зв'язки між даними різних таблиць та здійснювати операції з'єднання.

– Оптимізація запитів: Фізична модель також враховує способи оптимізації запитів до бази даних. Це може включати використання індексів, кешування, розподіл даних та інші стратегії для покращення продуктивності системи.

Фізична модель даних є ключовим етапом у розробці інформаційної системи, оскільки вона визначає, як дані будуть організовані та доступні для операцій збереження, оновлення та вибірки. Врахування вимог ефективності та оптимізації даних у фізичній моделі допомагає створити потужну та швидку систему обробки даних.

Для віртуальної примірки одягу інфологічна модель даних може включати наступні сутності та взаємозв'язки:

1. Сутність "Користувач" – ця сутність містить інформацію про користувачів системи, таку як їх ім'я, прізвище, адреса електронної пошти, пароль тощо.

2. Сутність "Одяг" – ця сутність містить інформацію про різні типи одягу, такі як футболки, штани, плаття тощо. Кожен тип одягу може мати свої унікальні атрибути, такі як колір, розмір, матеріал тощо.

3. Сутність "Модель" – ця сутність містить інформацію про моделі одягу, такі як певна футболка або плаття, і містить посилання на відповідний тип одягу.

4. Сутність "Фітинг" – ця сутність містить інформацію про параметри тіла користувача, такі як розмір грудей, талії та стегон.

5. Сутність "Замовлення" – ця сутність містить інформацію про замовлення, такі як дату замовлення, стан замовлення, вартість тощо.

6. Взаємозв'язок "Користувач – Замовлення" – цей взаємозв'язок показує, що кожен користувач може здійснювати багато замовлень, а кожне замовлення належить до одного користувача.

7. Взаємозв'язок "Замовлення – Модель" – цей взаємозв'язок показує, що кожне замовлення містить посилання на відповідну модель одягу.

8. Взаємозв'язок "Модель – Одяг" – цей взаємозв'язок показує, що кожна модель одягу належить до певного типу одягу.

9. Взаємозв'язок "Замовлення – Фітинг" – взаємозв'язок показує, що кожне замовлення містить інформацію про параметри тіла користувача, які були використані для вибору відповідної моделі одягу.

10. Взаємозв'язок "Користувач – Фітинг" – цей взаємозв'язок показує, що кожен користувач може мати різні параметри тіла, але кожен параметр тіла належить до одного користувача.

Така інфологічна модель даних може бути корисною при розробці віртуальної примірки одягу, оскільки вона дозволяє організовувати та зберігати різну інформацію про типи одягу, моделі, параметри тіла користувачів та замовлення. Ця модель також допоможе забезпечити ефективну інтеракцію між користувачами та системою віртуальної примірки одягу.

UML-діаграма використовується для моделювання більш загальних аспектів системи, включаючи її структуру та функціональність. UML-діаграма, що відображає користувацький досвід показано на рисунку 2.7.

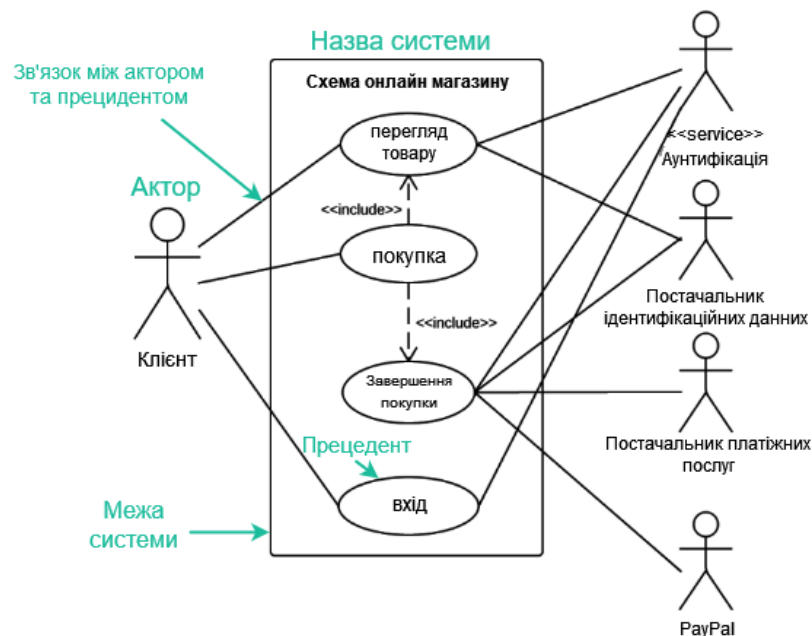


Рисунок 2.7 – Use-case діаграма

Під час розробки інформаційної системи після визначення інфологічної моделі даних та фізичної моделі, важливо враховувати кілька ключових аспектів, зокрема безпеку даних і особливості конкретної області, для якої система розробляється.

– **Безпека даних:** Під час розробки інформаційної системи, необхідно забезпечити високий рівень безпеки даних. Це означає, що необхідно розробити механізми для захисту інформації від несанкціонованого доступу, втрати, крадіжки або пошкодження. Зазвичай це включає в себе використання шифрування, контроль доступу на рівні користувача, аудит дій користувачів та застосування сучасних стандартів безпеки.

– **Вимоги до області:** Кожна інформаційна система специфічна для певної галузі або області. Під час розробки системи необхідно детально вивчити особливості цієї області та врахувати їх у проектуванні та функціональності системи. Наприклад, для системи управління медичною клінікою можуть бути специфічні вимоги до зберігання медичної інформації, виконання нормативних вимог до медичного забезпечення та регулярного оновлення лікарських даних.

– **Ефективність і масштабованість:** Під час розробки системи важливо враховувати потреби користувачів і забезпечити ефективну роботу системи. Це означає оптимізацію швидкості обробки даних, мінімізацію часу відгуку та забезпечення масштабованості системи для забезпечення росту та збільшення обсягу даних з часом.

– **Інтеграція з існуючими системами:** У багатьох випадках нова інформаційна система повинна інтегруватися з вже існуючими системами або сервісами. Під час розробки необхідно враховувати ці вимоги і забезпечити сумісність та взаємодію з існуючими системами.

– **Тестування та валідація:** Необхідно проводити систематичні тести та валідацію розробленої інформаційної системи. Це дозволить виявити та виправити помилки, забезпечити стабільну та надійну роботу системи перед її впровадженням.

– Правильне дотримання вимог стандартів безпеки даних та урахування особливостей конкретної області допоможуть забезпечити успішний розвиток та функціонування інформаційної системи, яка відповідає потребам користувачів та забезпечує захист даних.

Окрім того, під час проектування інформаційної системи важливо брати до уваги майбутні потреби системи та забезпечити її масштабованість, щоб у майбутньому система могла легко розширюватися та вдосконалюватися.

2.4.2 Проектування інтерфейсу інформаційної системи

Проектування інтерфейсу для віртуальної примірки одягу вимагає врахування кількох ключових факторів, що забезпечують ефективність та зручність користування. Деякі з них наведені нижче:

1. Інтуїтивно зрозумілий інтерфейс – це інтерфейс, який легко зрозуміти та використовувати користувачам, навіть якщо вони не мають попереднього досвіду використання системи або подібних до неї. Для того, щоб створити інтуїтивно зрозумілий інтерфейс, дизайнер повинен дотримуватися кількох ключових принципів:

Простота та логічність. Інтерфейс повинен бути простим та логічно збудованим, щоб користувач міг швидко зрозуміти, які дії необхідно виконувати, щоб досягти потрібного результату.

Інформативність. Інтерфейс повинен надавати користувачам достатньо інформації про те, що відбувається, щоб вони могли приймати обґрунтовані рішення та коректно виконувати дії.

Консистентність. Інтерфейс повинен мати однаковий стиль та поведінку на всіх сторінках та вікнах, щоб користувачі могли легко зрозуміти, як взаємодіяти з системою.

Відповідність очікуванням користувачів. Інтерфейс повинен відповідати очікуванням користувачів та їхнім потребам, щоб вони могли швидко та ефективно виконувати потрібні дії.

Наявність допомоги та підказок. Інтерфейс повинен мати допоміжну інформацію та підказки для користувачів, щоб вони могли швидко розібратися з системою та виконувати необхідні дії.

Загалом, створення інтуїтивно зрозумілого інтерфейсу є важливим елементом розробки будь-якої інформаційної систем

2. Візуалізація: Система повинна надавати можливість віртуальної примірки з використанням 3D-моделювання, для того щоб, дозволити користувачам оцінювати, як одяг буде виглядає на них.

3. Керування розмірами. Одним із важливих факторів, який впливає на комфортність носіння одягу – це є його розмір. Тому важливо, щоб система віртуальної примірки одягу мала можливість цю можливість, щоб користувачі могли точно налаштувати віртуальну примірку на свої реальні розміри.

Для досягнення цієї мети, система може мати вбудовані інструменти для зміни розмірів моделі, такі як масштабування, зміна довжини рукавів, штанин, поясу тощо. Користувачі можуть користуватися цими інструментами, щоб точно налаштувати віртуальну примірку під свої реальні розміри.

Крім того, система може надавати детальну інформацію про розміри одягу, таку як довжина рукава, штанин, обхват грудей тощо, щоб користувачі могли зрозуміти, який розмір одягу відповідає їх реальним розмірам. Це може бути важливим фактором при покупці онлайн, коли не існує можливості перевірити одяг перед покупкою.

Таким чином, можливість керування розмірами віртуальної примірки та надання інформації про розміри одягу є важливими функціями, які допоможуть користувачам отримати максимально точну інформацію про те, як одяг виглядає на них та як він підходить їх реальним розмірам.

– Надання інформації про товар: Користувачі повинні мати можливість переглядати детальну інформацію про продукт, таку як склад тканини, розміри та додаткові деталі, щоб допомогти їм прийняти рішення про покупку.

– Можливість вибору з різних стилів та колекцій одягу є важливим елементом в будь-якій системі віртуальної примірки одягу. Це дає користувачам

можливість знайти одяг, який відповідає їхньому стилю та персональним уподобанням. Наприклад, користувачі можуть вибрати одяг з різних колекцій, таких як весняна, літня, осіння та зимова, або з різних категорій, таких як спортивний, класичний, повсякденний та вечірній.

– Опції оплати та доставки: Користувачі повинні мати можливість вибрати з різних способів оплати та доставки, щоб зробити процес покупки максимально зручним.

– Підтримка клієнтів: Інтерфейс повинен мати можливість надати користувачам підтримку в разі потреби, зокрема, чат або електронну пошту для запитів щодо продуктів, замовлень та інших питань.

– Оптимізація швидкості: Інтерфейс повинен бути оптимізований для швидкості та продуктивності, щоб користувачі могли швидко переходити між сторінками та виконувати дії.

– Крос-платформна підтримка: Інтерфейс повинен бути розроблений для підтримки різних платформ та пристроїв, таких як: комп'ютери, планшети та мобільні пристрої, для того щоб забезпечити максимальну доступність користувачів.

– Тестування та вдосконалення: Інтерфейс повинен пройти тестування з метою виявлення та виправлення можливих помилок та проблем. При необхідності інтерфейс повинен вдосконалюватися та доповнюватися новими функціями.

У процесі проектування інтерфейсу для віртуальної примірки одягу, слід враховувати вимоги та потреби користувачів для того щоб, забезпечити максимальну зручність та ефективність взаємодії з системою, дати максимальну доступність користувачам на різних пристроях та платформах.

2.5 Висновки до розділу 2

У розділі 2 обрано спосіб віртуальної примірки одягу з використанням генеративних нейронних мереж. Встановлено, що формалізація та оброблення цифрового зображення включають такі кроки, як попередня обробка даних, аналіз

зображення. Ці процеси допомагають врахувати особливості одягу та забезпечити якість генерації.

У розділі описано процес проектування архітектури генеративної нейронної мережі для віртуальної примірки одягу. Обрано відповідні типи мереж, встановлено розміри шарів, функції активації та інші параметри, необхідні для ефективної генерації віртуального одягу на основі вхідного зображення.

Також у розділі наведено опис проектування інформаційної системи, яка ґрунтується на генеративній нейронній мережі для віртуальної примірки одягу. Визначено сутності та взаємозв'язки між ними, які необхідні для зберігання та обробки даних про вхідні зображення, віртуальний одяг та результати примірки. Проведено проектування інтерфейсу інформаційної системи для зручного взаємодії користувача з системою. Визначено основні елементи інтерфейсу, такі як вхідні поля, кнопки, панелі результатів та інші, які допомагають користувачу використовувати систему для віртуальної примірки одягу. Ці етапи є важливими для реалізації ефективної та зручної системи віртуальної примірки одягу на основі генеративних нейронних мереж.

Розділ 3 Програмна реалізація інформаційної системи з використанням генеративної нейронної мережі

3.1 Структура та функціональне призначення програмних складових інформаційної системи

В даній програмі використано модуль `argparse`. Він дозволяє розбирати аргументи командного рядка програми і створювати зручні інтерфейси командного рядка для користувача.

Цей модуль дозволяє зробити наступне:

- генерувати автоматичну документацію для програми на основі визначених аргументів командного рядка;
- перевіряти введені користувачем аргументи і повідомляти про помилки введення;
- забезпечити зручний інтерфейс командного рядка для користувача при запуску програми.

Модуль `'argparse'` дозволяє визначати аргументи, які програма очікує, встановлювати їх значення за замовчуванням, задавати обов'язкові аргументи, надавати довідкову інформацію та автоматично генерувати повний довідковий текст програми. Він також здатний перевіряти правильність переданих аргументів та повідомляти про помилки, якщо такі виникають.

Ще використання модуля `'argparse'` включає такі кроки:

1. Створення об'єкта `'ArgumentParser'`, який буде відповідати за обробку аргументів командного рядка.
2. Визначення аргументів, які програма може приймати, за допомогою методу `'add_argument()'`. Для кожного аргумента вказується його назва, тип, значення за замовчуванням, необхідність тощо.
3. Обробка командного рядка за допомогою методу `'parse_args()'`, який повертає об'єкт, що містить значення переданих аргументів.
4. Використання отриманих значень аргументів у програмі згідно зі своїми потребами.

Модуль `argparse` також дозволяє генерувати довідкову інформацію автоматично на основі визначених аргументів, що полегшує користувачам розуміння доступних опцій та використання програми.

Використання `argparse` у коді дозволяє програмі отримувати значення аргументів з командного рядка. Завдяки цьому, програма стає більш гнучкою та конфігурованою, оскільки користувач може вказувати значення аргументів при кожному запуску. Взаємодія функції `argparse` показана на рисунку 3.1.

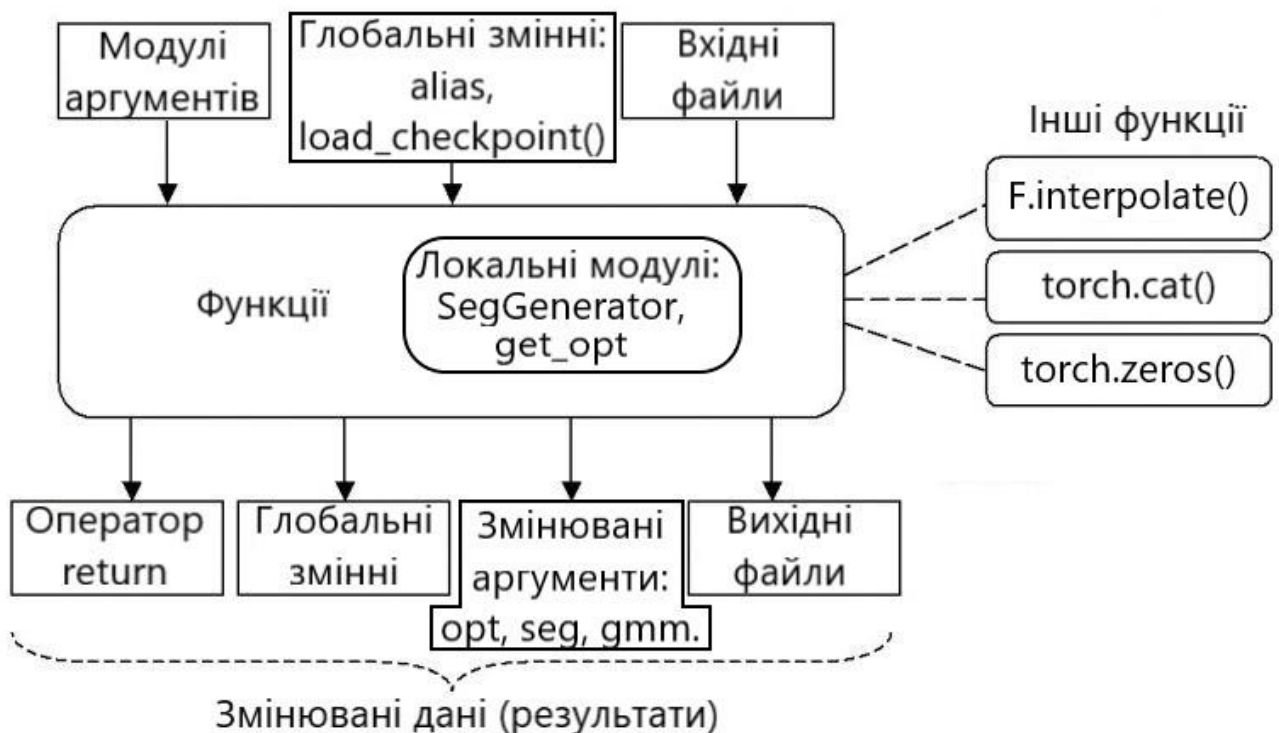


Рисунок 3.1 – Взаємодія функції `argparse` із зовнішніми частинами програми

Далі потрібно визначити клас `VITONDataset`, який є підкласом `torch.utils.data.Dataset` і використати для завантаження даних для задачі сегментації зображень та генерації одягу. Основна функціональність класу включає:

1. Завантаження даних: Конструктор класу приймає параметри `opt`, які містять налаштування для завантаження даних, такі як розмір зображень, шлях до даних тощо. Для завантаження даних використовуються шляхи до зображень,

файлів розмітки та масок. Дані нормалізуються та перетворюються в тензори за допомогою об'єкта `transforms.Compose`.

2. Обробка даних: Клас містить ряд методів, таких як `get_parse_agnostic`, `get_img_agnostic`, які застосовують обробку до зображень та масок для отримання агностичних зображень, які використовуються для генерації сегментації одягу. Для обробки використовуються різні операції, такі як маскування, розмальовування, зміщення точок тощо.

3. Завантаження одного елемента: Метод `__getitem__` викликається для завантаження даних одного елемента за вказаним індексом. Він завантажує зображення одягу, маску одягу, зображення постави, зображення сегментації, оброблює їх і повертає у вигляді словника.

4. Загальна кількість елементів: Метод `__len__` повертає загальну кількість елементів у наборі даних.

Крім того, код також визначає клас `VITONDataLoader`, який обгортає об'єкт `VITONDataset` і забезпечує ітерацію по пакетах даних.

Далі виконується обробка зображення за допомогою нейромережі. По-перше, розмір вхідного зображення зменшується до розміру (256, 192) за допомогою методу інтерполяції білінійного типу. Це зменшення розміру дозволяє прискорити обчислення та зменшити обсяг пам'яті, не втрачаючи важливих деталей.

Потім визначається кілька класів для архітектури нейронної мережі, яка використовується для семантичного сегментації та геометричного вирівнювання завдань. Ось короткий опис кожного класу:

1. `BaseNetwork`: Цей базовий клас надає загальні функціональні можливості для модулів нейронної мережі. Він містить методи для виведення архітектури мережі та ініціалізації ваг.

2. `SegGenerator`: Цей клас успадковується від `BaseNetwork` і представляє мережу генерації сегментації. Він приймає вхідне зображення та генерує карти сегментації. Він складається з кількох згорткових та розмірковувальних шарів.

Схему семантичної сегментації подано на рисунку 3.2.

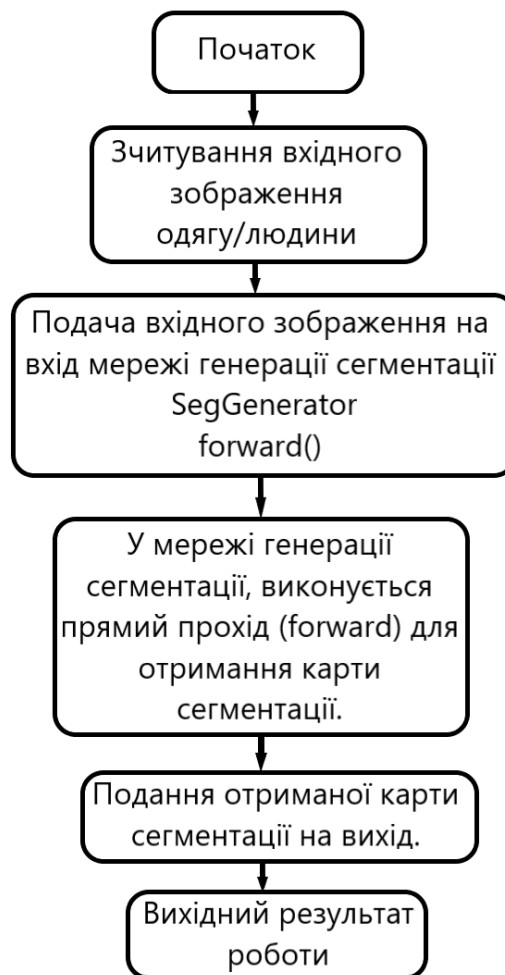


Рисунок 3.2 – Схема семантичної сегментації\

3. 'FeatureExtraction': Цей клас успадковується від 'BaseNetwork' і представляє мережу вилучення ознак. Він приймає вхідне зображення та вилучає високорівневі ознаки з нього. Він складається з кількох згорткових шарів.

4. 'FeatureCorrelation': Цей клас є простим модулем, який обчислює кореляцію ознак між двома наборами карт ознак.

5. 'FeatureRegression': Цей клас представляє мережу регресії ознак. Він приймає карту ознак як вхід та здійснює регресію параметрів перетворення.

6. 'TpsGridGen': Цей клас представляє генератор сітки тонкопластинчатого сплайна (TPS). Він генерує сітку контрольних точок для перетворення TPS.

У кожного класу є власний метод `forward`, який визначає потік обчислень через мережу. Метод `init_weights` ініціалізує ваги мережі за допомогою різних методів ініціалізації. Метод `print_network` виводить інформацію про архітектуру мережі та кількість параметрів.

Загалом ці класи визначають компоненти нейронної мережі, яка використовується для семантичної сегментації та геометричного вирівнювання завдань.

Потім виконується конкатенація кількох тензорів, що представляють різні аспекти зображення. Зокрема, використовуються зменшені версії карти сегментації, маски кольору, безпосереднього зображення, позиції та генерованого шуму. Ці тензори об'єднуються у новий тензор під назвою `seg_input`.

Потім, `seg_input` передається через модель сегментації (`seg`), яка генерує карти сегментації для різних класів. Отримані карти сегментації піддаються процесу `upsampling`, а потім обробляються функцією `argmax` для отримання кінцевої карти сегментації (`parse_pred`).

Далі, створюється тензор `parse_old` розміру `(batch_size, 13, load_height, load_width)`, заповнений нулями. Застосовується операція `scatter_`, яка заповнює тензор `parse_old` значеннями 1.0 згідно індексів, отриманих з карти сегментації `parse_pred`. Це дозволяє перетворити одноканальну карту сегментації в багатоканальну карту, де кожен канал відповідає певному класу.

Нарешті, створюється тензор `parse` розміру `(batch_size, 7, load_height, load_width)`, заповнений нулями. У циклі для кожного класу виконується обчислення піксельних сум карти `parse_old` за відповідними індексами з "

`labels`". Це дозволяє створити окремий канал для кожного класу в тензорі `parse`.

Отриманий тензор `parse` є кінцевим результатом обробки зображення за допомогою нейромережі. Він містить інформацію про присутність різних об'єктів та класифікує їх згідно зі зазначеними мітками.

Модуль сегментації генерації зображено на рисунку 3.3.



Рисунок 3.3 – Модуль сегментації генерації

Наступним кроком виконується генерація відзеркаленого зображення за допомогою генеративно-суперечливої мережі (GMM). Починаємо з інтерполяції декількох тензорів зображення до розміру (256, 192) за допомогою методу "nearest" (найближчий сусід).

Тензори "img_agnostic", "parse" (вибираємо лише другий канал), "pose" і "c" зменшуються до вказаного розміру за допомогою функції "F.interpolate". Ці зменшені тензори об'єднуються в один тензор "gmm_input" шляхом конкатенації.

Далі, викликається функція "gmm" з переданим "gmm_input" та "c_gmm" в якості вхідних параметрів. Це викликає процес генерації відзеркаленого зображення за допомогою GMM. Результатом є сітка "warped_grid", яка містить перетворені координати для відображення пікселів з вхідного зображення на вихідне.

Використовуючи функцію "F.grid_sample", зображення "c" та "cm" зміщуються згідно отриманої сітки "warped_grid". Результати зберігаються в тензорах "warped_c" та "warped_cm".

Таким чином, отримується відзеркалене зображення "warped_c" та відповідна маска кольору "warped_cm" на основі вхідних даних і генеративно-суперечливої мережі. Модуль деформації одягу показано на рисунку 3.4.

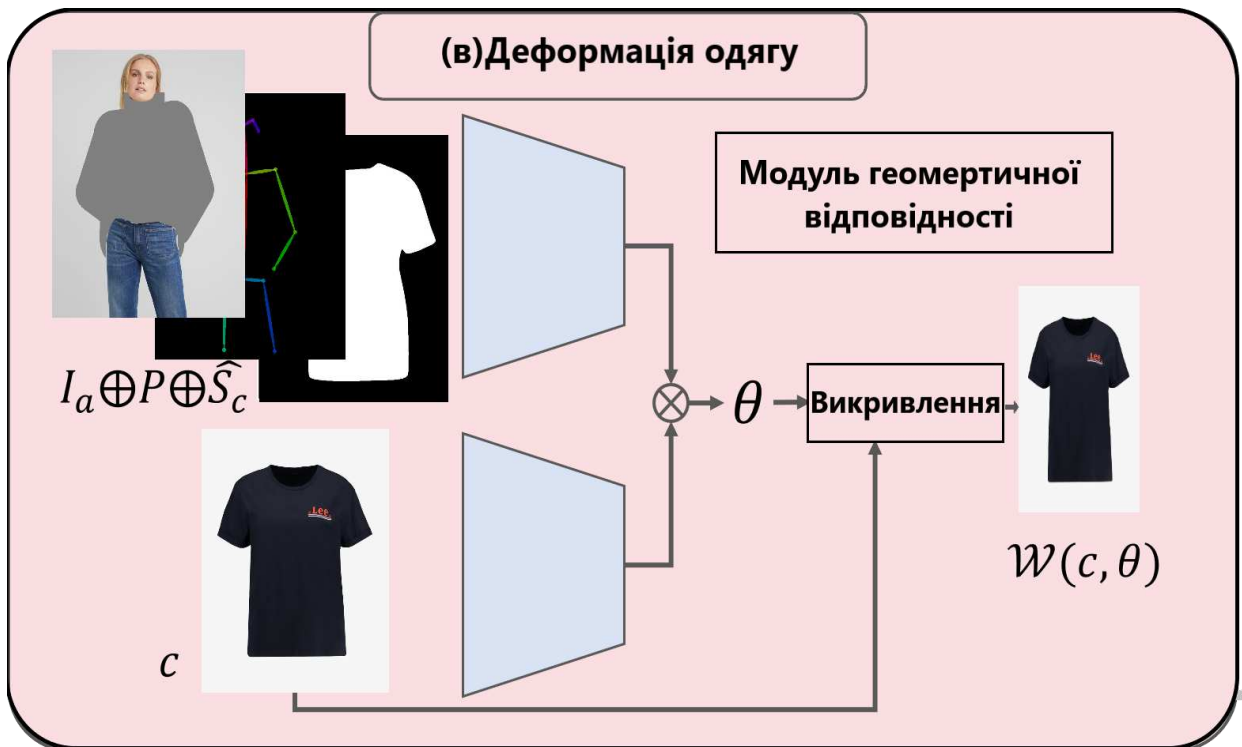


Рисунок 3.4 – Модуль деформації одягу

Останнім кроком виконується додаткова обробка результатів синтезу "try-on" та збереження отриманих зображень.

Спочатку створюється маска "misalign_mask" шляхом віднімання картки кольору "warped_cm" від другого каналу карти сегментації "parse". Якщо значення у масці "misalign_mask" менше нуля, то воно замінюється на нуль.

Далі, створюється тензор "parse_div", який об'єднує карту сегментації "parse" та маску "misalign_mask". Також відбувається віднімання значень з маски "misalign_mask" від другого каналу "parse_div" для вирівнювання місця злиття між одягом та тілом особи.

Потім викликається функція "alias" з переданими параметрами: зображення "img_agnostic", позицію "pose", відзеркалене зображення одягу "warped_c", карта сегментації "parse" та "parse_div", а також маска

"misalign_mask". Ця функція відповідає за обробку та об'єднання вхідних даних для отримання кінцевого результату "output".

Далі, генеруються назви зображень для збереження. Кожне зображення має пару – ім'я зображення та відповідне ім'я одягу. Отримані назви зображень зберігаються у змінній "unpaired_names".

Зображення "output" разом із відповідними назвами "unpaired_names" зберігаються у вказаній директорії "opt.save_dir" за допомогою функції "save_images".

У разі, якщо поточний крок ($i + 1$) ділиться на "opt.display_freq" без залишку, виводиться повідомлення про поточний крок обробки.

Таким чином, в цьому коді виконується фінальна обробка результатів синтезу "try-on" та збереження отриманих зображень. Модуль пробного синтезу виконано на рисунку 3.5.

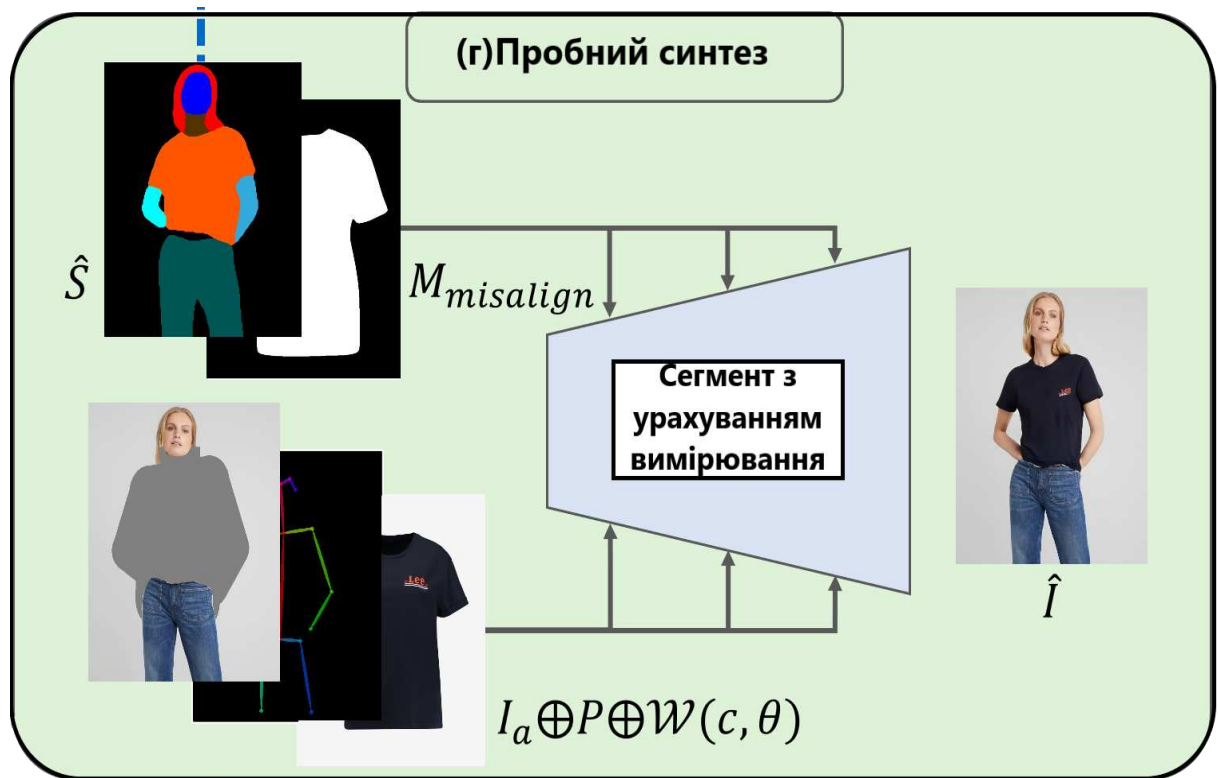


Рисунок 3.5 – Модуль пробного синтезу

Далі описаний код, який допомагає з управлінням шумом, збереженням зображень і завантаженням точок контролю моделі у для тестування та оцінки моделі віртуальної примірки.

1. Функція `gen_noise(shape)`, генерує шум розміру `shape`. Вона створює масив нулів типу `np.uint8`, заповнює його випадковим шумом за допомогою функції `cv2.randn()`, нормалізує його до діапазону `[0, 1]` і повертає його у вигляді `torch.tensor` типу `torch.float32`.

2. Функція `save_images(img_tensors, img_names, save_dir)`, зберігає тензори зображень `img_tensors` зі списком імен `img_names` у заданому каталозі `save_dir`. Вона перетворює тензори в масиви `NumPy`, виконує необхідні операції з масивами для коректного збереження зображень (нормалізація, конвертація типів тощо) і зберігає зображення у форматі `JPEG` за допомогою модуля `PIL`.

3. Функція `load_checkpoint(model, checkpoint_path)` завантажує збережену точку контролю моделі `model` з вказаного шляху `checkpoint_path`. Вона перевіряє наявність файлу з точкою контролю, завантажує його за допомогою `torch.load()`, і

передає стан моделі методу `load_state_dict()` для відновлення моделі з точкою контролю.

3.2 Особливості реалізації програмних складових інформаційної системи

Інтерфейс для програми я зробив з допомогою `tkinter` – це стандартний модуль мови Python, він надає інтерфейс для створення графічного користувацького інтерфейсу (GUI) і його показано на рисунку 3.6.

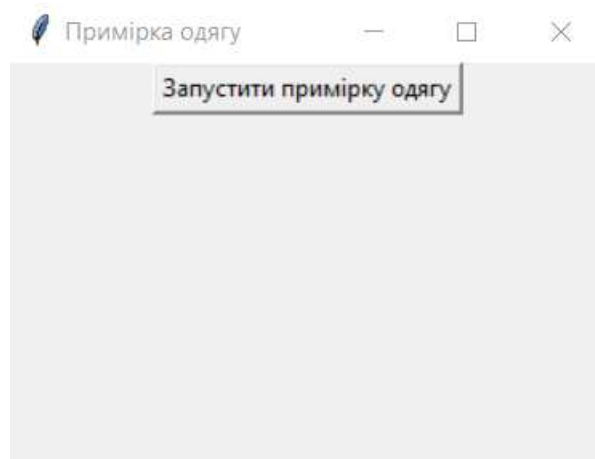


Рисунок 3.6 – Головний інтерфейс запуску програми

Далі показаний результат виконання примірки програми на рисунку 3.7.

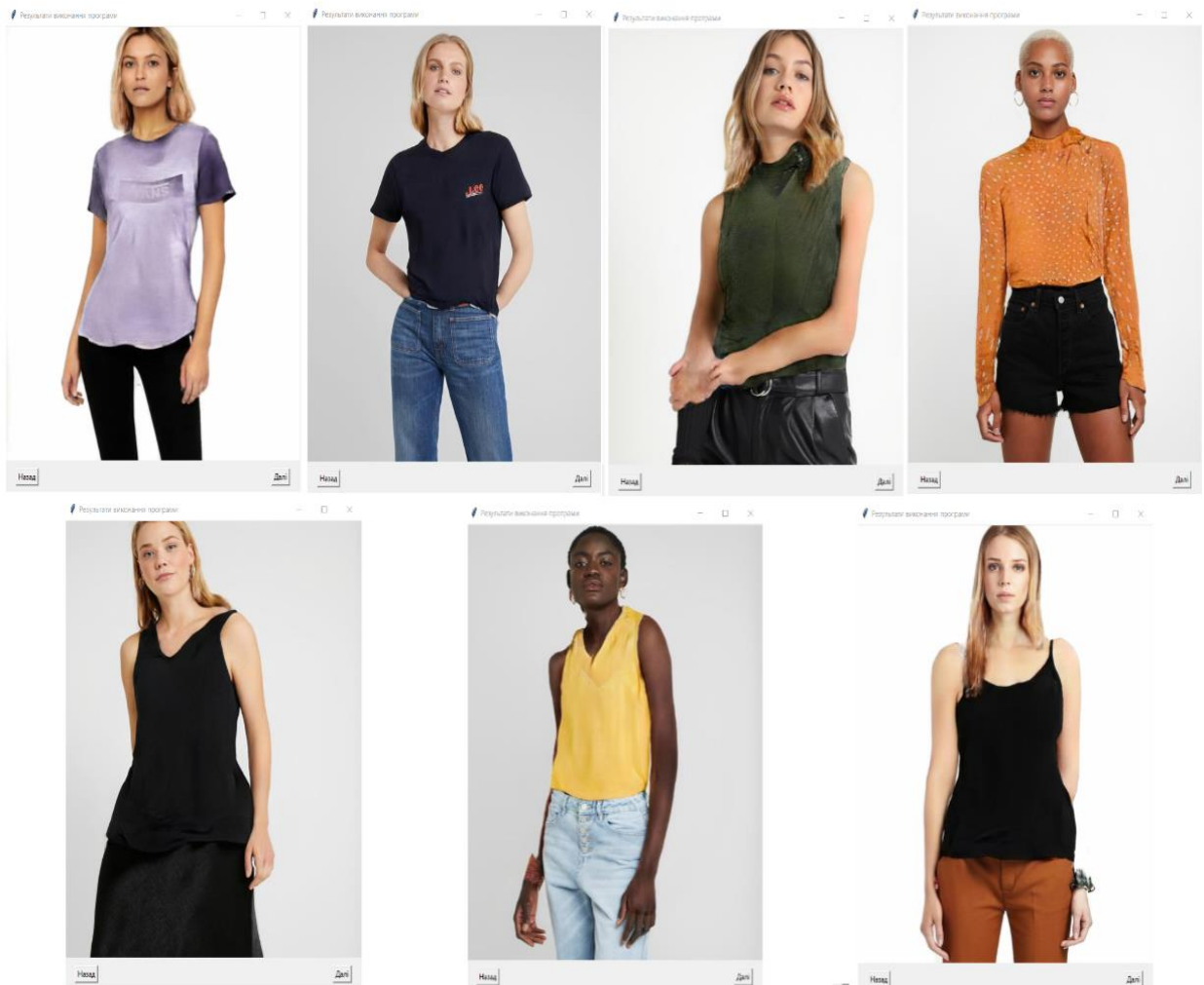


Рисунок 3.7 – Результат роботи програми

Далі показані кнопки для перегляду результату зображені на рисунку 3.8.

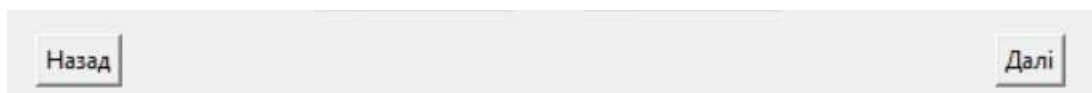


Рисунок 3.8 – Кнопки для перегляду результату програми

Створюються дві кнопки для перемикання зображень: "Назад" і "Далі".

Перша кнопка "Назад" створюється за допомогою `tk.Button`. Вона має текст "Назад" і при натисканні на неї виконується функція `self.show_previous_image`. Потім кнопка розміщується на головному вікні за допомогою `pack(side=tk.LEFT, padx=20, pady=10)`, що вказує на розташування кнопки зліва від інших елементів і задає відступи по горизонталі і вертикалі.

Друга кнопка "Далі" створюється аналогічно першій кнопці, з текстом "Далі" і функцією `self.show_next_image`, яка виконується при натисканні на кнопку. Кнопка розміщується справа від інших елементів за допомогою `pack(side=tk.RIGHT, padx=20, pady=10)`.

Отже, для перемикання зображень: "Назад" і "Далі", потрібно прив'язати відповідні функції, які виконуються при натисканні на кнопки.

Далі потрібно описати метод. Дві функції `show_previous_image` і `show_next_image`, які відповідають за перемикання на попереднє і наступне зображення в колекції.

У функції `show_previous_image` виконується наступне:

1. Змінна `self.current_image_index` оновлюється шляхом віднімання 1 від поточного значення та використання оператора `%` для забезпечення циклічного перемикання на попереднє зображення у випадку, якщо досягнуто початку колекції.

2. Викликається функція `self.show_image()`, яка відповідає за відображення поточного зображення.

У функції `show_next_image` виконується наступне:

1. Змінна `self.current_image_index` оновлюється шляхом додавання 1 до поточного значення та використання оператора `%` для забезпечення циклічного перемикання на наступне зображення у випадку, якщо досягнуто кінця колекції.

2. Викликається функція `self.show_image()`, яка відповідає за відображення поточного зображення.

Отже, метод описує функціонал перемикання на попереднє і наступне зображення в колекції шляхом оновлення змінної `self.current_image_index` і виклику відповідної функції для відображення поточного зображення.



Рисунок 3.9 – Результат роботи програми з різними типами одягу

Далі розглянемо результати примірки з одним типом одягу, але з різними людьми. Він показаний на рисунку 3.10.

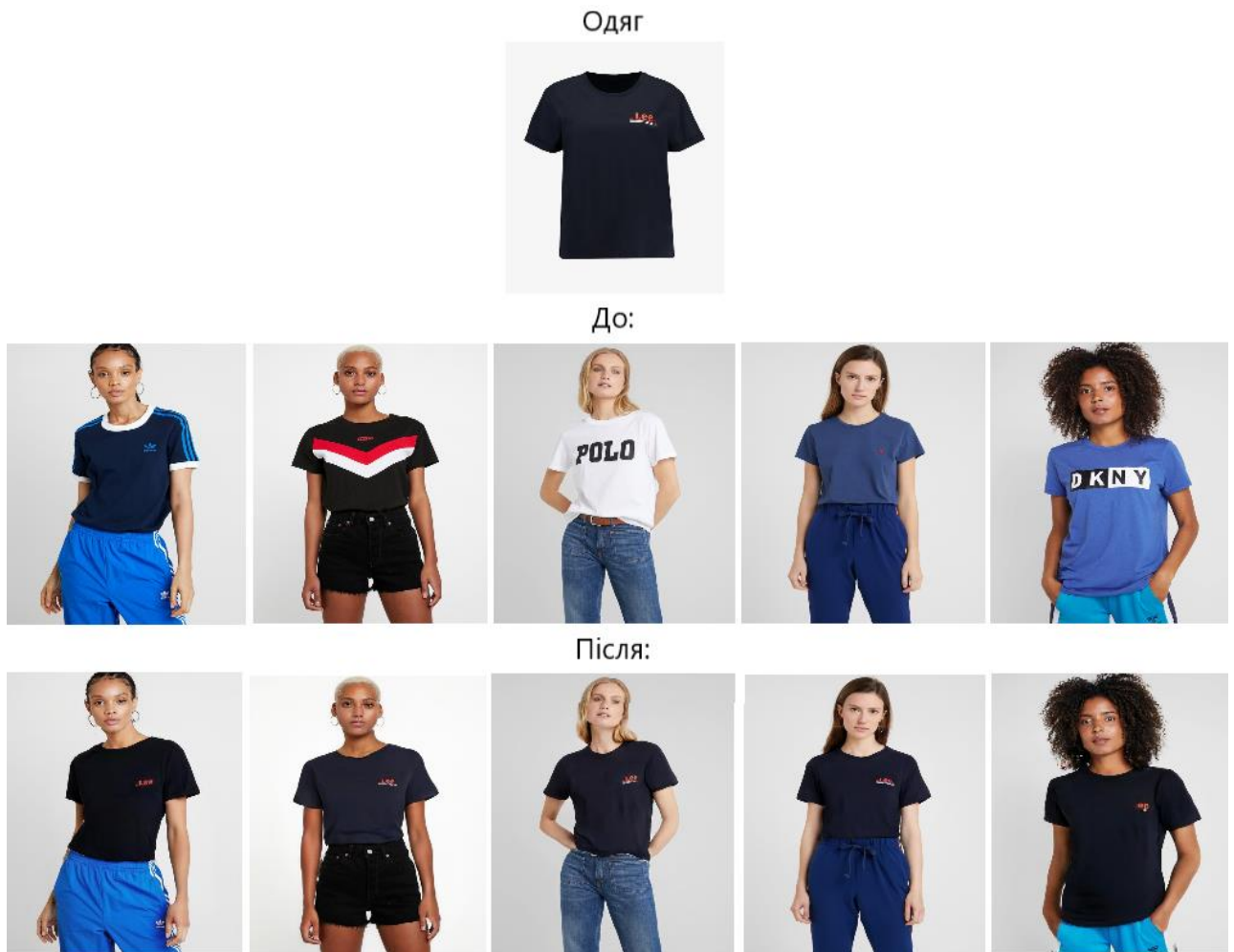


Рисунок 3.10 – Примірка одягу на різних людях

3.3 Експериментальне тестування інформаційної системи

Тестування інформаційної системи є важливим етапом в будь-якій розробці.

Для тестування можна взяти попередні результати програми з різними типами одягу. Його відображено на рисунку 3.11.



Рисунок 3.11 – Погано перенесені малюнки на футболках

На даному рисунку видно, що примірка самого одягу пройшла успішно, але малюнки на футболках перенеслися погано.

Далі можна взяти приклади роботи програми з некоректною приміркою його зображено на рисунку 3.12.

Оригінальні фото

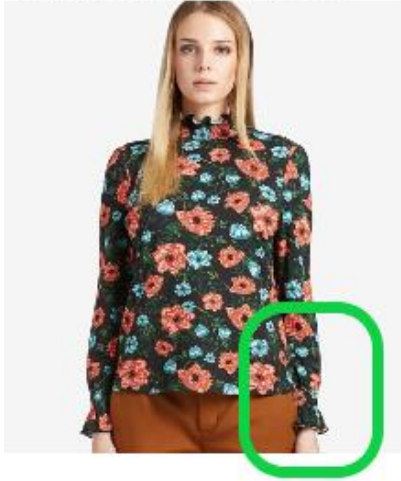


Фото одягу



+

=

Результат примірки



+

=



+

=

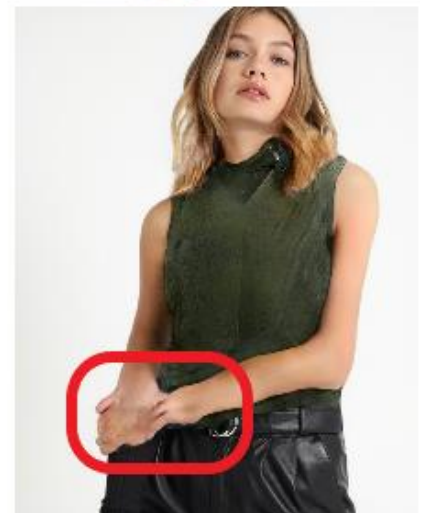


Рисунок 3.12 – Некоректна робота за деякими фото

Як бачимо, в результаті примірки програма некоректно перенесла руки людей.

Щоб запустити процес примірки потрібно ввести в термінал команду. Для того, щоб не робити це вручну

Потрібно виконати команду, яка запускає скрипт "test.py" у середовищі Python. Команда включає прапорець "--name" зі значенням "try_on". Функція `run_command()` використовує модуль `subprocess` для виконання команди.

Загальний синтаксис команди `subprocess.run()` передбачає вказання команди як рядок із параметрами командного рядка, аргумент `shell=True` вказує на виконання команди у командному рядку операційної системи.

Отже, в даному випадку, команда `python test.py --name try_on` буде виконана, що може призвести до запуску скрипту "test.py" з аргументом `--name` і значенням "try_on" у середовищі Python.

3.4 Вимоги до розгортання інформаційної системи та інструкція користувача

Після запуску програми користувача зустрічає початковий інтерфейс в якому, ви можете розпочати примірku одягу. Інтерфейс показаний на рисунку 3.13.

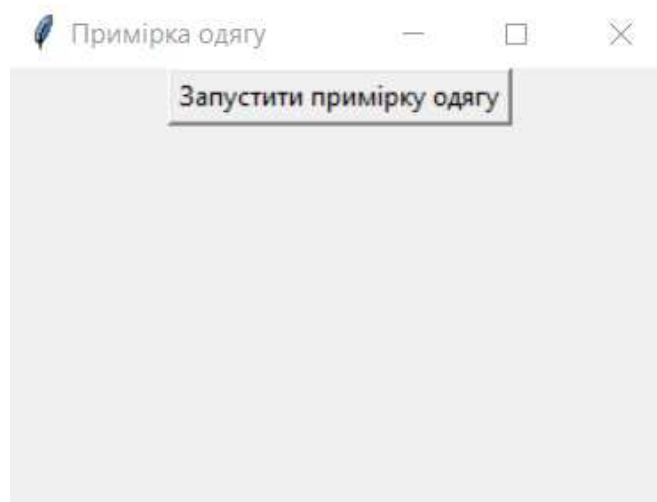


Рисунок 3.13 – Початковий інтерфейс програми

Після цього ви можете натиснути на кнопку «Запустити примірку одягу». Далі програма починає примірку. Її показано на рисунку 3.15.

```

PS D:\Viton\VITON-HD> & C:/Users/Admin/AppData/Local/Programs/Python/Python38/python.exe d:/Viton/VITON-HD/tryon1.py
Namespace(alias_checkpoint='alias_final.pth', batch_size=1, checkpoint_dir='./checkpoints', dataset_dir='./datasets', dataset_list='test_pairs.txt', dataset_mode='test', display_freq=1, gmm_checkpoint='gmm_final.pth', grid_size=5, init_type='xavier', init_variance=0.02, load_height=1024, load_width=768, name='try_on', ngf=64, norm_G='spectralaliasinstance', num_upsampling_layers='most', save_dir='./results', seg_checkpoint='seg_final.pth', semantic_nc=13, shuffle=False, workers=1)
Network [SegGenerator] was created. Total number of parameters: 34.5 million. To see the architecture, do print(network).
Network [ALIASGenerator] was created. Total number of parameters: 100.5 million. To see the architecture, do print(network).
C:\Users\Admin\AppData\Local\Programs\Python\Python38\lib\site-packages\torch\nn\functional.py:4236: UserWarning: Default grid_sample and affine_grid behavior has changed to align_corners=False since 1.3.0. Please specify align_corners=True if the old behavior is desired. See the documentation of grid_sample for details.
  warnings.warn(
step: 1
step: 2
step: 3
step: 4
step: 5
step: 6
step: 7

```

Рисунок 3.14 – Процес роботи програми

Весь результат зберігається в папці. Місце знаходження результату можна змінити на іншу папку, вказавши інше ім'я в даному коді, який показано на рисунку 3.16.

```

command = "python test.py --name try_on"

```

Рисунок 3.15 – Команда для запуску програми та зі створенням назви папки

Вікно зі шляхом до результату, відкривається у коді на рисунку 3.17.

```

app = ImageGallery(root, directory="./results/try_on", width=550, height=650)

```

Рисунок 3.16 – Шлях до результату у директорії: «./results/try_on»

Після завершення роботи програми відкривається вікно з результатами. Він показаний на рисунку 3.18.



Рисунок 3.17 – Вікно з результатом програми

Для того щоб переглянути всі результати, можна скористатись кнопками навігації «Назад» та «Далі».

Для примірки інших результатів потрібно вказати назву фото людей і одягу у блокноті. Він показаний на рисунку. 3.19.

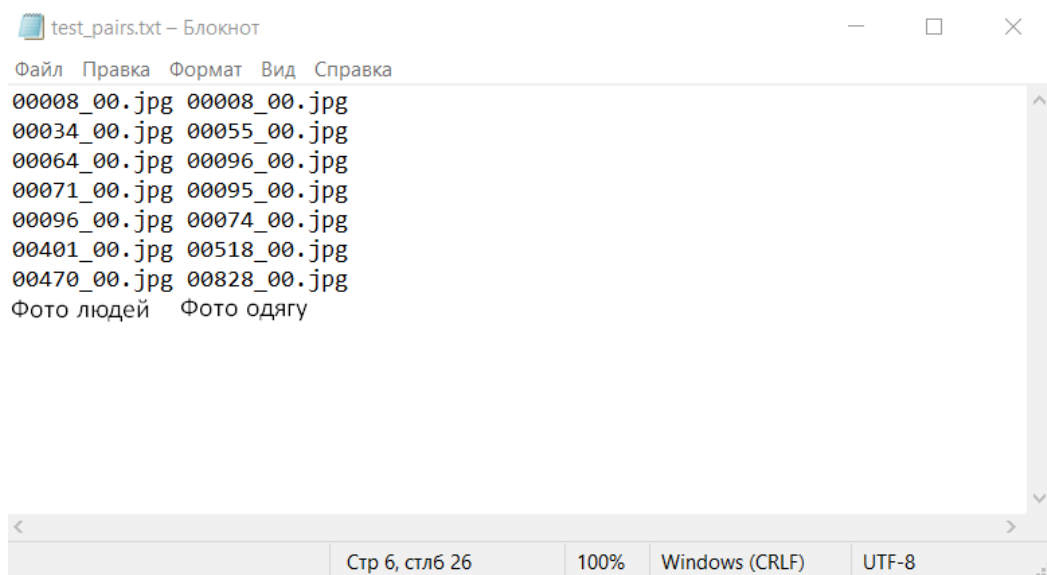


Рисунок 3.18 – Примірка одяжі до людей

Самі фотографії одягу і людей знаходяться у папці «datasets», «test», «cloth/image». Вони показані на рисунку 3.20

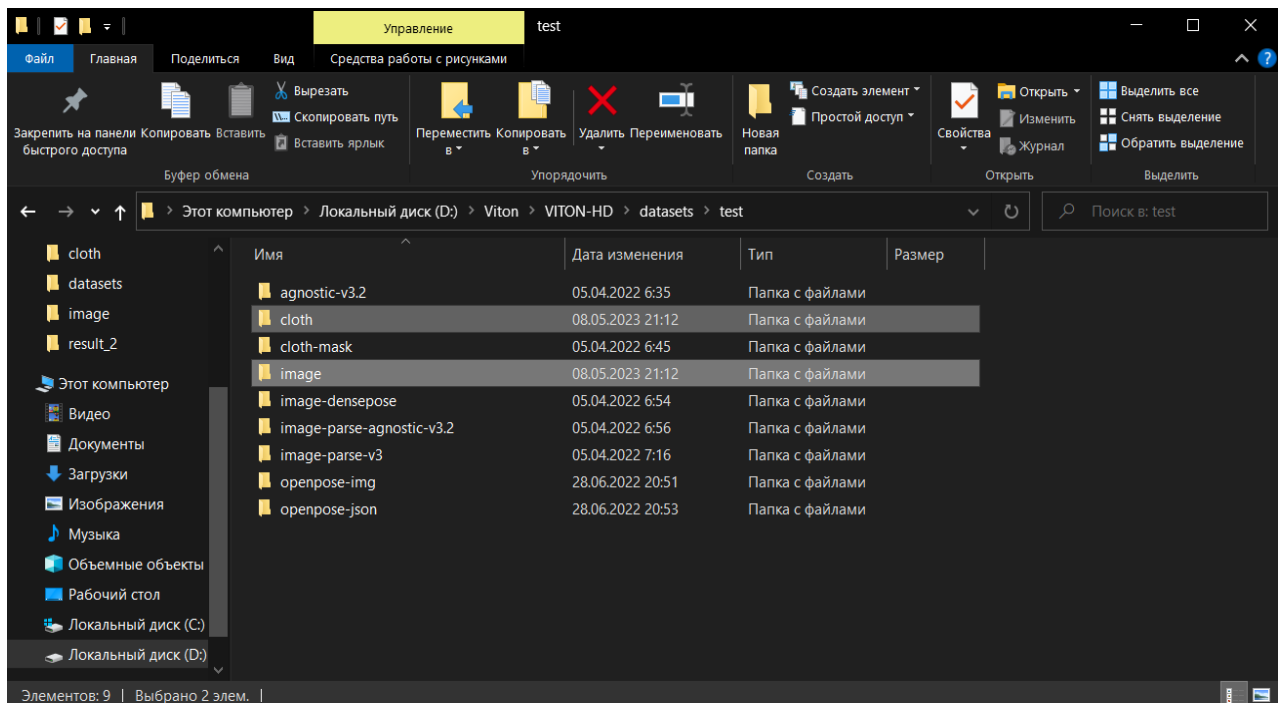


Рисунок 3.19 – Папки з фото для примірки одягу до людей

Для того щоб дана програма запустилась та стабільно працювала, потрібні такі системні та програмні вимоги:

- Visual Code версії 1.68.1 або Visual Studio 2019-2022;

- Python 3.8;
- операційна система Windows 10/11;
- процесор з частотою 2000 МГц;
- 7 гігабайт пам'яті дискового простору;
- 8 гігабайт оперативної пам'яті.

3.5 Висновки до розділу 3

У розділі 3 описано результати програмної реалізації інформаційної системи з використанням ГНМ. Структура системи складається з чотирьох основних компонентів: попередня обробка, генерація сегментації, деформація одягу та пробний синтез зображень.

У розділі описано особливості реалізації програмних складових інформаційної системи з використанням генеративної нейронної мережі. Технічні аспекти, такі як вибір технологій та бібліотек для реалізації мережі, оптимізація алгоритмів, обробка даних та налаштування параметрів системи, були також описані. Наведено методологію тестування, використовуваних метрик для оцінки якості та ефективності системи, проведення експериментів на реальних результатах.

Описано вимоги до розгортання інформаційної системи. Вказано необхідні апаратні та програмні засоби, налаштування середовища, залежності та інші вимоги для успішного встановлення та роботи системи. Також вказана та описана інструкція користувача, яка пояснює процес взаємодії з програмою.

Висновки

У результаті виконання кваліфікаційної роботи бакалавра було спроектовано та реалізовано інформаційну систему з використанням генеративної нейронної мережі.

У роботі обрано спосіб віртуальної примірки одягу на основі генеративної нейронної мережі. Спроектовано архітектуру нейронної мережі для виконання задачі віртуальної примірки одягу. Обрано відповідні типи мереж, встановлено розміри шарів, функції активації та інші параметри, необхідні для ефективної генерації віртуального одягу на основі вхідного зображення.

Проведено проектування інтерфейсу інформаційної системи для зручного взаємодії користувача з системою. Визначено основні елементи інтерфейсу, такі як вхідні поля, кнопки, панелі результатів та інші, які допомагають користувачу використовувати систему для віртуальної примірки одягу. Структура реалізованої системи має чотири основні компонент: попередня обробка, генерація сегментації, деформація одягу та пробний синтез зображень. Описано особливості реалізації програмних складових інформаційної системи з використанням генеративної нейронної мережі.

Наведено методологію тестування, використовуваних метрик для оцінки якості та ефективності системи, проведення експериментів на реальних результатах. В результаті проведення експериментальних тестувань встановлено, що розроблена інформаційна система якісно покращує процес віртуальної примірки одягу за цифровими зображеннями людини та предметів одягу.

Описано вимоги до розгортання інформаційної системи. Вказано необхідні апаратні та програмні засоби, налаштування середовища, залежності та інші вимоги для успішного встановлення та роботи системи. Також вказана та описана інструкція користувача, яка пояснює процес взаємодії з програмою.

Перелік посилань

1. Технології вибору одягу і взуття та догляду за ними | Твоя бібліотека. History | Твоя електронна бібліотека. URL: <https://uahistory.co/pidruchniki/hodzicka-labor-training-8-class-2016/17.php>
2. Роль технологій доповненої та віртуальної реальності у післявоєнному відновленні українського бізнесу. *MC.today, Media for Creators*. URL: <https://mc.today/uk/blogs/rol-tehnologij-dopovnenoyi-ta-virtualnoyi-realnosti-u-pislyavoyennomu-vidnovlenni-ukrayinskogo-biznesu/>
3. Самолюк Т.А. Нейромережі GAN в створенні нових моделей. *Комп'ютерні засоби, мережі та системи*. 2019. № 18. С. 86-90.
4. Hu L. Design and implementation of a component-based intelligent clothing style cad system. *Computer-Aided Design and Applications*. 2020. Vol. 18, No. 1. Pp. 22-32.
5. Гуртовий Ю.В., Коріонов Д. Віртуальна примірка 3D моделі одягу на відео засобами машинного навчання. *Наукові записки молодих учених*. 2021. № 8. С. 1-8.
6. Silva E.S., Hassani H., Madsen D.Ø. Big data in fashion: transforming the retail sector. *Journal of Business Strategy*. 2019. Vol. 41, No. 4. Pp. 21-27.
7. Бармак О.В., Радюк П.М., Молчанова М.О., Собко О.В. Підходи до практичного аналізу обчислювальних алгоритмів. *Вісник Хмельницького національного університету. Технічні науки*. 2022. № 303(6). С. 102-105.
8. Ghoreishi M., Harpponen A. The case of fabric and textile industry: The emerging role of digitalization, internet-of-things and industry 4.0 for circularity. In: Yang, X.S., Sherratt, S., Dey, N., Joshi, A. (eds) *Proceedings of Sixth International Congress on Information and Communication Technology*. Lecture Notes in Networks and Systems, Vol. 216. Singapore : Springer, Inc., 2021. Pp. 189-200.
9. Radiuk P.M., Mazurets O.V., Skrypnyk T.K., Moroz O.V. Intelligent data analysis using artificial neural networks for decision making in the education domain.

Herald of Khmelnytskyi National University. Technical sciences. 2021. Vol. 303, No 6. Pp. 111-114.

10. Drieiev O., Dorenskyi O., Drieieva H. Neural network method for detecting textural anomalies in a digital image. *Central Ukrainian Scientific Bulletin. Technical Sciences.* 2022. Vol. 2, No. 5(36). Pp. 335-346.

11. Pavlova O., Radiuk P., Kravchuk S., Kulbachnyi V. Information system for public places and institutions visualization with opportunities of inclusive access and optimal routing. *Computer systems and information technologies.* 2022. Vol. 1, No 6. Pp. 62-68.

12. Gollapudi S. OpenCV with Python. Learn computer vision using OpenCV: With deep learning CNNs and RNNs. Berkeley, CA : Apress, 2019. Pp. 31-50.

13. Radiuk P.M. Applying 3D U-Net architecture to the task of multi-organ segmentation in computed tomography. *Applied Computer Systems.* 2020. Vol. 25, No 1, Pp. 43-50.

14. A Review on generative adversarial networks: Algorithms, theory, and applications / J. Gui et al. *IEEE Transactions on Knowledge and Data Engineering.* 2022. Vol. 35, No. 4. Pp. 3313-3332.

15. Radiuk P.M., Hrypynska N.V. A framework for exploring and modeling neural architecture search methods. *The 4th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2020) : CEUR-Workshop Proceedings.* Vol. 2604. (Lviv, Ukraine, 23-24 April 2020). CEUR-WS.org, Aachen, 2020. Pp. 1060-1074.

16. Radiuk P., Pavlova O., Avsiyevych V., Kovalenko V. Convolutional neural network for parking slots detection. *The 3rd International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS-2022) : CEUR-Workshop Proceedings.* Vol. 3156. (Khmelnyskyi, Ukraine, 23-25 March 2022). CEUR-WS.org, Aachen, 2022. Pp. 284-293.

17. Virtual Dressing Room Application Using GANs / A. Bhagyalakshmi et al. *2023 9th International Conference on Advanced Computing and Communication*

Systems (ICACCS-2023) : Proceedings. (Coimbatore, India, 17-18 March 2023). IEEE, Inc., 2023.

18. Chen L., Fang Q., Chen Y. Intelligent clothing interaction design and evaluation system based on DCGAN image generation module and Kansei engineering. *Journal of Physics: Conference Series*. 2021. Vol. 1790, No. 1. Pp. 012025.

19. Oeldorf C., Spanakis G. LoGANv2: Conditional style-based logo generation with generative adversarial networks. *2019 18th IEEE International Conference on Machine Learning And Applications (ICMLA-2019)* : Proceedings. (Boca Raton, FL, USA, 16–19 December 2019). IEEE, Inc., 2019. Pp. 462-468.

20. Choi I., Park S., Park J. Generating and modifying high resolution fashion model image using StyleGAN. *2022 13th International Conference on Information and Communication Technology Convergence (ICTC-2022)* : Proceedings. (Jeju Island, Republic of Korea, 19–21 October 2022). IEEE, Inc., 2022.

21. You can try without visiting: a comprehensive survey on virtually try-on outfits / H. Ghodh bani et al. *Multimedia Tools and Applications*. 2022. Vol. 81. Pp. 19967-1998.

22. Дишкант В.О. Оптимізація та побудова системи генерування зображень на основі генеративно-змагальних мереж : магістерська дис. : 122 Комп'ютерні науки / Дишкант Владислав Олегович. Київ, 2022. 83 с.

23. JUST LAUNCHED | Looklet Dressing Room. *Looklet | The most powerful solution for your fashion e-com imagery*. 2023. URL: <https://www.looklet.com/looklet-dressing-room>

24. REVERY | Create your virtual dressing room. *Revery.ai*. 2023. URL: <https://www.revery.ai/>

25. VIRTUAL TRY-ON WORKING IN A CALL WITH US. *Zyler | Virtual Clothing Try-On*. 2023. URL: <https://www.zyler.com/get-started>

26. Analysis of deep learning methods in adaptation to the small data problem solving / I. Krak et al. *Intellectual Systems of Decision Making and Problem of Computational Intelligence (ISDMCI-2022)* : Lecture Notes in Data Engineering,

Computational Intelligence, and Decision Making. 2022. Vol. 149. (Zalizniy Port, Ukraine, 23-27 May 2022). Springer, Cham. Pp. 333-352.

27. Mohammadi S. O., Kalhor A. Smart Fashion: A review of ai applications in virtual try-on & fashion synthesis. *Journal of Artificial Intelligence and Capsule Networks*. 2021. Vol. 3, No. 4. Pp. 284-304.

28. PyTorch: An imperative style, high-performance deep learning library / A. Paszke et al. *Advances in Neural Information Processing Systems* : Proceedings. (Vancouver, BC, Canada, December 8-14, 2019). 2019. Vol. 32. Curran Associates, Inc., 2019. Pp. 8024-8035.

29. Python 3.8 | Python is a programming language that lets you work quickly and integrate systems more effectively. *Python.org*. URL: <https://www.python.org/>

30. torchvision – Torchvision 0.15 documentation. *PyTorch.org*. URL: <https://pytorch.org/vision/stable/index.html>

ДОДАТКИ

Додаток А

Лістинг програмного коду

Модуль інтерфейсу програми:

```

import os
from PIL import Image, ImageTk
import tkinter as tk
import subprocess

class ImageGallery:
    def __init__(self, master, directory, width=550, height=650):
        self.master = master
        self.directory = directory
        self.width = width
        self.height = height

        self.images = self.load_images()
        self.current_image_index = 0

        #фрейм для відображення зображення
        self.image_frame = tk.Frame(master, width=width, height=height)
        self.image_frame.pack()

        #Label для відображення зображення
        self.image_label = tk.Label(self.image_frame)
        self.image_label.pack()

        #кнопки для перемикання між зображеннями
        self.previous_button = tk.Button(master, text="Назад", command=self.show_previous_image)
        self.previous_button.pack(side=tk.LEFT, padx=20, pady=10)

        self.next_button = tk.Button(master, text="Далі", command=self.show_next_image)
        self.next_button.pack(side=tk.RIGHT, padx=20, pady=10)

        self.show_image()

    def load_images(self):
        images = []
        for filename in os.listdir(self.directory):
            if filename.endswith('.jpg') or filename.endswith('.png'):
                images.append(Image.open(os.path.join(self.directory, filename)))
        return images

    def show_image(self):
        #поточне зображення
        current_image = self.images[self.current_image_index].resize((self.width, self.height), Image.ANTIALIAS)
        photo = ImageTk.PhotoImage(current_image)
        self.image_label.configure(image=photo)
        self.image_label.image = photo

    def show_previous_image(self):
        #перемикання на попереднє зображення
        self.current_image_index = (self.current_image_index - 1) % len(self.images)
        self.show_image()

    def show_next_image(self):
        #перемикання на наступне зображення
        self.current_image_index = (self.current_image_index + 1) % len(self.images)
        self.show_image()

    def run_command():
        command = "python test.py --name try_on"
        subprocess.run(command, shell=True)
        #створення вікна
        root = tk.Tk()

```

```

root.geometry('300x200')

#створення кнопки з текстом
button = tk.Button(root, text="Запустити примірку одягу", command=run_command)

#відображаємо кнопку на екрані
button.pack()

#головний цикл програми
root.title("Примірка одягу")
root.mainloop()
root = tk.Tk()
app = ImageGallery(root, directory="./results/try_on", width=550, height=650)
root.title("Результати виконання програми")
root.mainloop()

```

Модуль реалізації генеративної нейронної мережі для віртуальної примірки одягу:

```

import argparse
import os
import torch
from torch import nn
from torch.nn import functional as F
import torchgeometry as tgm
from datasets import VITONDataset, VITONDataLoader
from networks import SegGenerator, GMM, ALIASGenerator
from utils import gen_noise, load_checkpoint, save_images

def get_opt():
    parser = argparse.ArgumentParser()
    parser.add_argument('--name', type=str, required=True)
    parser.add_argument('-b', '--batch_size', type=int, default=1)
    parser.add_argument('-j', '--workers', type=int, default=1)
    parser.add_argument('--load_height', type=int, default=1024)
    parser.add_argument('--load_width', type=int, default=768)
    parser.add_argument('--shuffle', action='store_true')
    parser.add_argument('--dataset_dir', type=str, default='./datasets/')
    parser.add_argument('--dataset_mode', type=str, default='test')
    parser.add_argument('--dataset_list', type=str, default='test_pairs1.txt')
    parser.add_argument('--checkpoint_dir', type=str, default='./checkpoints/')
    parser.add_argument('--save_dir', type=str, default='./results/')
    parser.add_argument('--display_freq', type=int, default=1)
    parser.add_argument('--seg_checkpoint', type=str, default='seg_final.pth')
    parser.add_argument('--gmm_checkpoint', type=str, default='gmm_final.pth')
    parser.add_argument('--alias_checkpoint', type=str, default='alias_final.pth')
    # common
    parser.add_argument('--semantic_nc', type=int, default=13, help='# of human-parsing map classes')
    parser.add_argument('--init_type', choices=['normal', 'xavier', 'xavier_uniform', 'kaiming',
        'orthogonal', 'none'], default='xavier')
    parser.add_argument('--init_variance', type=float, default=0.02, help='variance of the initialization distribution')
    # for GMM
    parser.add_argument('--grid_size', type=int, default=5)
    # for ALIASGenerator
    parser.add_argument('--norm_G', type=str, default='spectralaliasinstance')
    parser.add_argument('--ngf', type=int, default=64, help='# of generator filters in the first conv layer')
    parser.add_argument('--num_upsampling_layers', choices=['normal', 'more', 'most'], default='most',
        help='If \'more\', add upsampling layer between the two middle resnet blocks. '
        'If \'most\', also add one more (upsampling + resnet) layer at the end of the generator.')
    opt = parser.parse_args()
    return opt

def test(opt, seg, gmm, alias):
    up = nn.Upsample(size=(opt.load_height, opt.load_width), mode='bilinear')
    gauss = tgm.image.GaussianBlur((15, 15), (3, 3))

```

```

# gauss.cuda()
test_dataset = VITONDataset(opt)
test_loader = VITONDataLoader(opt, test_dataset)
with torch.no_grad():
    for i, inputs in enumerate(test_loader.data_loader):
        img_names = inputs['img_name']
        c_names = inputs['c_name']['unpaired']
        ...

        coda with cuda
        img_agnostic = inputs['img_agnostic'].cuda()
        parse_agnostic = inputs['parse_agnostic'].cuda()
        pose = inputs['pose'].cuda()
        c = inputs['cloth']['unpaired'].cuda()
        cm = inputs['cloth_mask']['unpaired'].cuda()
        ...

        img_agnostic = inputs['img_agnostic']
        parse_agnostic = inputs['parse_agnostic']
        pose = inputs['pose']
        c = inputs['cloth']['unpaired']
        cm = inputs['cloth_mask']['unpaired']

# 1. Сегментація генерації
parse_agnostic_down = F.interpolate(parse_agnostic, size=(256, 192), mode='bilinear')
pose_down = F.interpolate(pose, size=(256, 192), mode='bilinear')
c_masked_down = F.interpolate(c * cm, size=(256, 192), mode='bilinear')
cm_down = F.interpolate(cm, size=(256, 192), mode='bilinear')
seg_input = torch.cat((cm_down, c_masked_down, parse_agnostic_down, pose_down,
                       gen_noise(cm_down.size())), dim=1)
parse_pred_down = seg(seg_input)
parse_pred = gauss(up(parse_pred_down))
parse_pred = parse_pred.argmax(dim=1)[:, None]

parse_old = torch.zeros(parse_pred.size(0), 13, opt.load_height, opt.load_width, dtype=torch.float)
parse_old.scatter_(1, parse_pred, 1.0)

labels = {
    0: ['background', [0]],
    1: ['paste', [2, 4, 7, 8, 9, 10, 11]],
    2: ['upper', [3]],
    3: ['hair', [1]],
    4: ['left_arm', [5]],
    5: ['right_arm', [6]],
    6: ['noise', [12]]
}
parse = torch.zeros(parse_pred.size(0), 7, opt.load_height, opt.load_width, dtype=torch.float)
for j in range(len(labels)):
    for label in labels[j][1]:
        parse[:, j] += parse_old[:, label]

# 2. Деформація одягу
agnostic_gmm = F.interpolate(img_agnostic, size=(256, 192), mode='nearest')
parse_cloth_gmm = F.interpolate(parse[:, 2:3], size=(256, 192), mode='nearest')
pose_gmm = F.interpolate(pose, size=(256, 192), mode='nearest')
c_gmm = F.interpolate(c, size=(256, 192), mode='nearest')
gmm_input = torch.cat((parse_cloth_gmm, pose_gmm, agnostic_gmm), dim=1)
_, warped_grid = gmm(gmm_input, c_gmm)
warped_c = F.grid_sample(c, warped_grid, padding_mode='border')
warped_cm = F.grid_sample(cm, warped_grid, padding_mode='border')

# 3. Пробний синтез
misalign_mask = parse[:, 2:3] - warped_cm
misalign_mask[misalign_mask < 0.0] = 0.0
parse_div = torch.cat((parse, misalign_mask), dim=1)
parse_div[:, 2:3] -= misalign_mask
output = alias(torch.cat((img_agnostic, pose, warped_c), dim=1), parse, parse_div, misalign_mask)
unpaired_names = []
for img_name, c_name in zip(img_names, c_names):
    unpaired_names.append('{}_{}'.format(img_name.split('_')[0], c_name))
save_images(output, unpaired_names, os.path.join(opt.save_dir, opt.name))
if (i + 1) % opt.display_freq == 0:
    print("step: {}".format(i + 1))
def main():

```

```
opt = get_opt()
print(opt)
if not os.path.exists(os.path.join(opt.save_dir, opt.name)):
    os.makedirs(os.path.join(opt.save_dir, opt.name))
seg = SegGenerator(opt, input_nc=opt.semantic_nc + 8, output_nc=opt.semantic_nc)
gmm = GMM(opt, inputA_nc=7, inputB_nc=3)
opt.semantic_nc = 7
alias = ALIASGenerator(opt, input_nc=9)
opt.semantic_nc = 13
load_checkpoint(seg, os.path.join(opt.checkpoint_dir, opt.seg_checkpoint))
load_checkpoint(gmm, os.path.join(opt.checkpoint_dir, opt.gmm_checkpoint))
load_checkpoint(alias, os.path.join(opt.checkpoint_dir, opt.alias_checkpoint))
seg.eval()
gmm.eval()
alias.eval()
test(opt, seg, gmm, alias)
if __name__ == '__main__':
    main()
```

Додаток Б

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

СПОСІБ ВІРТУАЛЬНОЇ ПРИМІРКИ ОДЯГУ З ВИКОРИСТАННЯМ ГЕНЕРАТИВНО НЕЙРОННИХ МЕРЕЖ

Виконав:

студент 4 курсу, групи КН-19-1
Оксанюк Максим Сергійович

Керівник:

старший викладач кафедри КН
Радюк Павло Михайлович

Актуальність

Віртуальна примірка одягу за допомогою цифрових зображень набуває все більшої популярності та значущості в сучасному світі. Застосування штучного інтелекту для генерації цифрових зображень та обробки даних дозволяє створювати реалістичні та точні віртуальні моделі одягу, що має великий потенціал у різних сферах, таких як мода, електронна комерція, дизайн одягу та інше.

Створення інформаційної системи, яка забезпечує віртуальну примірку одягу, дозволить користувачам зручно та точно оцінювати вигляд різних предметів одягу на собі без необхідності фізичного приміряння. Це може полегшити процес вибору одягу, зекономити час та зусилля, а також допомогти виробникам та дизайнерам у покращенні своїх продуктів та послуг.

Таким чином, розробка інформаційної системи для віртуальної примірки одягу є актуальною темою, яка поєднує передові технології штучного інтелекту, комп'ютерного зору та моделювання, впливає на розвиток модної індустрії та сприяє зручності та ефективності в процесі вибору та покупки одягу.

Мета і задачі роботи

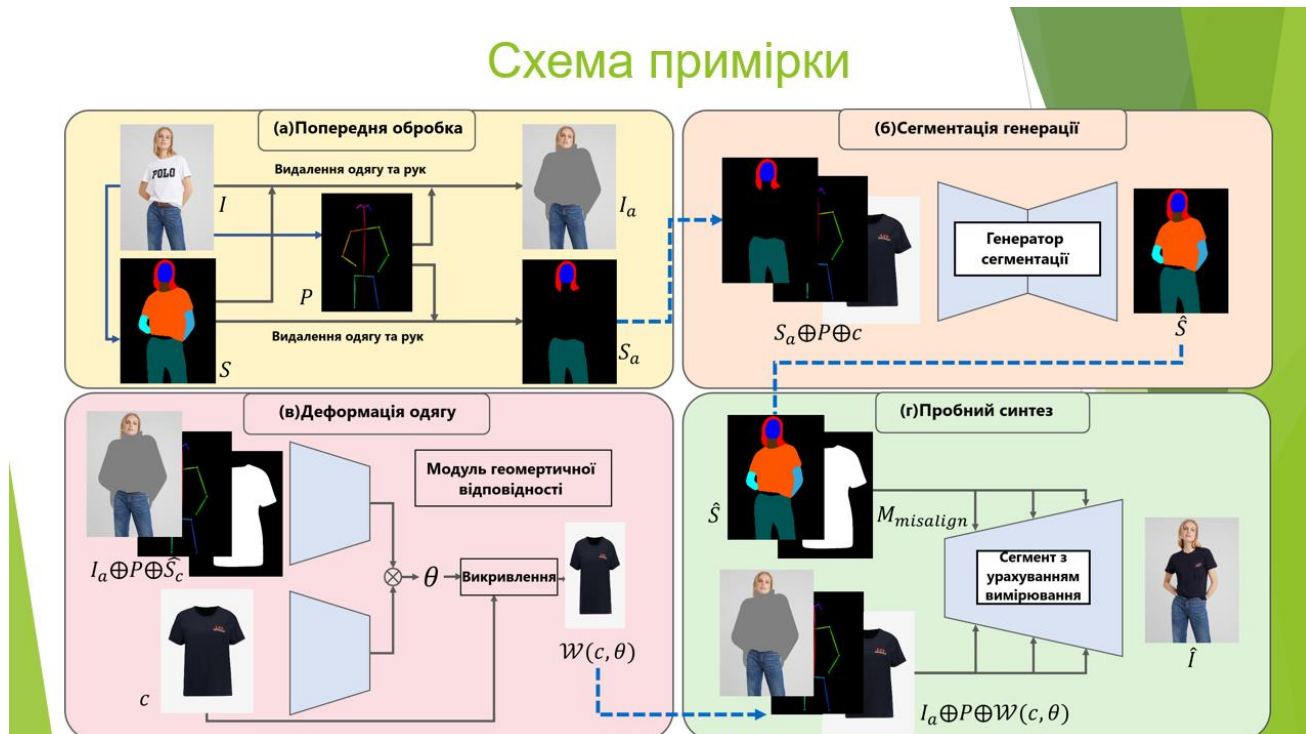
Метою кваліфікаційної роботи бакалавра є покращення віртуальної примірки одягу за цифровими зображеннями людини та предметів одягу для цього потрібно зробити наступні завдання.

1. Провести аналіз засобів штучного інтелекту для оброблення та генерування цифрових зображень та обрати найкращий.
2. Використати обраний засіб оброблення та генерування цифрових зображень для розв'язання задачі віртуальної примірки за цифровими зображеннями людини та предметів одягу.
3. Реалізувати спосіб віртуальної примірки одягу у вигляді модуля програмного забезпечення.
4. Провести експериментальне тестування реалізованого модуля за еталонними наборами даних.

Основні кроки

1. Збір даних
2. Передобробка даних
3. Створення генератора та дискримінатора: Генеративна нейронна мережа (GAN) складається з двох основних компонентів: генератора та дискримінатора.
4. Навчання генеративної нейронної мережі.
5. Оцінка результатів.

Схема примірки



Примірка різного одягу на одній людині



Примірка одного виду одягу на різних людях

Одяг



До:



Після:



Висновок

Під час виконання кваліфікаційної роботи бакалавра було розроблено та реалізовано спосіб віртуальної примірки одягу з використанням генеративних нейронних мереж. Основною метою роботи було створення системи, яка дозволяє користувачам віртуально приміряти різні види одягу без фізичного носіння.

У процесі дослідження були використані сучасні методи та алгоритми глибокого навчання для розв'язання поставленої задачі. Було проведено аналіз та вибір найбільш підходящих моделей та архітектур для розробки системи віртуальної примірки одягу

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 7%

ID: 114154 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА Додано в БД: 2023-05-27 Автора: М.С. Оксанюк Керівники: П.М. Радюк Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	64235	994	1150 (2%)	21 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:
Кафедра КН

ID перевірки:
1015286929

Дата перевірки:
27.05.2023 18:32:06 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
27.05.2023 18:34:59 EEST

ID користувача:
100005671

Назва документа: КН-19-1 Оксанюк

Кількість сторінок: 62 Кількість слів: 10708 Кількість символів: 80967 Розмір файлу: 2.36 MB ID файлу: 1014959353

5.82% Схожість

Найбільша схожість: 2.21% з джерелом з Бібліотеки (ID файлу: 1014959355)

4.91% Джерела з Інтернету 505 Сторінка 64

2.91% Джерела з Бібліотеки 96 Сторінка 66

0.16% Цитат

Цитати 1 Сторінка 67

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Спосіб віртуальної примірки одягу з використанням генеративних нейронних мереж

Автор: студент групи КН-19-1 Оксаною Максим Сергійович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: док. філ., ст. викл. Радюк П.М.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<i>відповідає</i>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі Оксаною М.С., не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти програмного коду, що не мають авторства і містять поширені конструкції; поміж запозичень знаходяться загальновідомі терміни та скорочення.

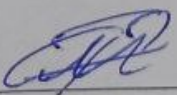
Обсяг запозичень, визначений системами виявлення збігів/ ідентичності/схожості, складає:

- за системою Anti-Plagiarism: 1.0%;

- за системою Unichек: 5.82 %.

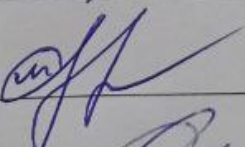
Отже, запозичення є допустимими та відносяться до описаних вище і адресуються до першоджерел, що, з урахуванням наведених обґрунтувань, свідчить на користь кваліфікаційної роботи.

Керівник роботи



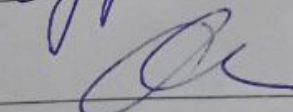
Павло РАДЮК

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



**ВІДГУК НАУКОВОГО КЕРІВНИКА
на кваліфікаційну роботу бакалавра**

студента гр. КН-19-1 Оксанюка Максима Сергійовича

за темою Спосіб віртуальної примірки одягу з використанням генеративних нейронних мереж

1. Актуальність теми

Реалізація одягу онлайн за допомогою цифрових засобів набуває все більшої популярності та значущості в світі. Застосування інформаційних технологій та штучного інтелекту до процесу примірки одягу дасть змогу користувачам зручно та точно оцінювати вигляд різних предметів одягу на собі без необхідності фізичного приміряння. Таким чином, розроблення нового програмного рішення, яке поєднає сучасні технології штучного інтелекту, комп'ютерного зору та моделювання для виконання віртуальної примірки одягу, є актуальною задачею.

**2. Відповідність роботи предметній області Стандарту спеціальності
122 Комп'ютерні науки**

Відповідно до стандарту бакалавра вищої освіти України спеціальності 122 – Комп'ютерні науки, описом предметної галузі, об'єктом та предметом вивчення є математичні, інформаційні та імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи й технології отримання, зберігання, обробки, передачі та використання інформації. Метою поданої роботи є покращення віртуальної примірки одягу за цифровими зображеннями людини та предметів одягу. Метою роботи досягнуто внаслідок використання методів, способів та алгоритмів розв'язання теоретичних і прикладних задач, що виникають у процесі проєктування та розроблення інформаційних технологій. Отже, результати виконання кваліфікаційної роботи відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

Під час виконання кваліфікаційної роботи студент Оксанюк Максим Сергійович проявив себе кваліфікованим фахівцем та дисциплінованим студентом, вчасно виконуючи поставлені перед ним завдання. Як у процесі розроблення прикладного програмного забезпечення, так і під час написання пояснювальної записки студент успішно засвоїв компетентності та результати навчання. Загалом Оксанюк М.С. опанував професійні навички та компетентності, що відповідають виконанню освітньо-професійної програми рівня вищої освіти «Бакалавр» за спеціальністю 122 – Комп'ютерні науки.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Результати роботи та їхня обґрунтована практична значущість одержані та обумовлені студентом особисто, як наслідок виконання ним усіх поставлених завдань.

5. Ступінь оволодіння методами дослідження

У процесі аналізу способу віртуальної примірки одягу з використанням генеративних нейронних мереж та проектування і розроблення на його основі інформаційної системи студент Оксанюк М.С. продемонстрував достатній рівень компетентностей та володіння необхідними інструментами й обладнанням, методами, методиками та технологіями галузі інформаційних технологій.

6. Повнота та якість розкриття теми роботи

Тема роботи повністю обґрунтована й розкрита; актуальність предметної галузі та відомі дослідження щодо обраної тематики проаналізовані достатньо. Усі поставлені завдання в роботі виконані повністю. Розроблене програмне забезпечення для валідації та верифікації спроектованої інформаційної системи відповідає технічним вимогам спеціальності 122 – Комп'ютерні науки.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Матеріал кваліфікаційної роботи Оксанюка М.С. подано логічно, послідовно, аргументовано та є таким, що відповідає поставленій меті. Мова і стиль викладення роботи відповідають стандартам, що забезпечує доступність сприймання матеріалу і відповідає вимогам до сучасних кваліфікаційних робіт.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Розроблене в роботі програмне забезпечення може бути використане в діяльності онлайн-магазинів для надання користувачам нового засобу для вибору одягу. Впровадження програмного забезпечення також дасть змогу виробникам та дизайнерам ефективніше створювати та реалізувати нові моделі одягу.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

З огляду на рівень виконання та забезпечення всіх необхідних вимог, вважаю, що кваліфікаційна робота Оксанюка Максима Сергійовича може бути допущена до захисту. Рекомендована оцінка – «добре».

Керівник _____



_____ док. філ., ст. викл. каф. КН Павло РАДЮК



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-19-1 Оксанока Максима Сергійовича
за темою: Спосіб віртуальної примірки одягу з використанням генеративних нейронних мереж

1. Актуальність обраної теми

Віртуальна примірка одягу за допомогою цифрових зображень набуває все більшої популярності та значущості в сучасному світі. Застосування штучного інтелекту для генерації цифрових зображень та обробки даних дає змогу створювати реалістичні та точні віртуальні моделі одягу, що має великий потенціал у різних сферах, таких як мода, електронна комерція, дизайн одягу та інше.

2. Повнота розкриття мети та завдань роботи

У результаті виконання кваліфікаційної роботи бакалавра мету та завдання роботи розкрито повністю. Проведено глибокий аналіз предметної галузі, розглянуто та проаналізовано чимало літературних джерел щодо обраної тематики. Чітко визначено структуру інформаційної системи віртуальної примірки одягу з використанням генеративних нейронних мереж та реалізовано прикладне програмне забезпечення для проведення експериментального тестування над системою. У підсумку, мета та завдання поданої до захисту роботи є розкритими повністю.

3. Зміст кожного розділу роботи

У першому розділі кваліфікаційної роботи проведено аналіз галузі віртуальної примірки одягу за цифровими зображеннями людини та предметів одягу. Другий розділ присвячений проєктуванню способу віртуальної примірки одягу з використанням генеративних нейронних мереж. У третьому розділі наведено та описано програмну реалізацію інформаційної системи за спроектованим способом віртуальної примірки одягу. Насамкінець оформлено висновки до виконаних у роботі завдань.

4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблена автором інформаційна система ґрунтується на навченій заздалегідь генеративній нейронній мережі, що виконує віртуальну примірку одягу. Проведено експериментальне тестування за якісними метриками, в результаті чого встановлено, що мету роботи успішно досягнуто.

5. Якість оформлення кваліфікаційної роботи бакалавра

Робота відповідає вимогам науково-технічного стилю написання, має чітку структуру та включає всі необхідні розділи, як от, перелік скорочень, вступ, огляд літератури, методологія, результати експериментів, висновки та перелік посилань. Кожен розділ повністю виконує своє функціональне призначення і має чіткий зміст. Використані джерела належним чином процитовані й включені до списку використаної літератури, відповідно до наукових стандартів цитування. Використана автором структура та оформлення сприяють легкому сприйняттю та розумінню матеріалу.

6. Недоліки кваліфікаційної роботи бакалавра

Робота також містить недоліки. У роботі приділено недостатньо уваги оцінюванню інформаційної системи за статистичними показниками точності, влучності, повноти тощо. Не розкриті питання щодо виникнення візуальних артефактів на зображенні, а саме помилкове відтворення реалістичних деталей одягу та людини, візуальна оклюзія контурів одягу та неправильне відображення фізичних ознак моделі.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота

З огляду на рівень виконання та забезпечення всіх необхідних вимог вважаю, що подана кваліфікаційна робота бакалавра може бути допущена до захисту.

Рекомендована оцінка – « добре ».

Рецензент _____



доц., к. ф.-м. н. Надія Івченко