

VTON AI

A COURSE PROJECT REPORT

21CSE306P – APPLIED GENERATIVE AI

Submitted by

ROHAN KUPPILI [RA2211028010206]

CHIRANJEEV KUMAR [RA2211029010019]

SHREYANSH RAJESH KUMBHARE [RA2211003011495]

SWARNA GHANTY [RA2211026010333]

Under the Guidance of

Dr. SWATHY R.

Assistant Professor, Department of Networking and Communications

in partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING



SCHOOL OF COMPUTING

**DEPARTMENT OF NETWORKING AND
COMMUNICATIONS**

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

NOVEMBER 2024



SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Established in the year 1983

Department of Networking and Communications

SRM Institute of Science and Technology

Own Work Declaration Form

Degree/ Course : 21CSE306P – Applied Generative AI

Student Name : Rohan Kuppili, Chiranjeev Kumar,
Shreyansh Rajesh Kumbhare, Swarna Ghanty

Registration Number: RA2211028010206, RA2211028010019,
RA2211003011495, RA2211026010333

Title of Work : VTON AI

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

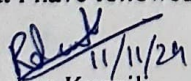
I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

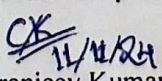
- Clearly referenced / listed all sources as appropriate.
- Referenced and put in inverted commas all quoted text (from books, web, etc).
- Given the sources of all pictures, data etc. that are not my own.
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present.
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources).
- Compiled with any other plagiarism criteria specified in the Course handbook/ University website.

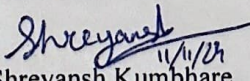
I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

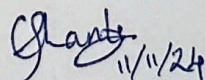
DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.


Rohan Kuppili
RA2211028010206


Chiranjeev Kumar
RA2211029010019


Shreyansh Kumbhare
RA2211003011495


Swarna Ghanty
RA2211026010333

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr.T.V. Gopal, and Dr. Revathi Venkataraman, Professor & Chairperson**, School of Computing, SRM Institute of Science and Technology, for their valuable support during the project course.

We are incredibly grateful to Head of the Department, **Dr. M. Lakshmi**, Professor and Head, Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, for offering us the project course: Applied Generative AI.

We want to convey our thanks to our Course Coordinator, **Dr. V. Vaishnavi Moorthy**, Assistant Professor, Industry expert, **Mr. Murali Meenakshi Sundaram** and our faculty mentor **Dr. R. Swathy**, Assistant Professor, Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, for their inputs and support during the project phase.

We express our respect and thanks to **Department HOD**, SRM Institute of Science and Technology, for providing us an opportunity to pursue this project course.

We register our immeasurable thanks to our **Faculty Advisor**, School of Computing, SRM Institute of Science and Technology, for leading and helping us to complete our course.

We sincerely thank the Networking and Communications department staff members, SRM Institute of Science and Technology, for their help during our project.

Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.



SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 21CSE306P-Applied Generative AI project report titled “VTON AI” is the bonafide work of “ROHAN KUPPILI [RA2211028010206], CHIRANJEEV KUMAR [RA2211029010019], SHREYANSH RAJESH KUMBHARE [RA2211003011495] AND SWARNA GHANTY [RA2211026010333]” who carried out the project work[internship] under our supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate.


SIGNATURE 11/11/24

Dr. R. SWATHY

Assistant Professor,
Department of Networking and
Communications





SIGNATURE

Dr. M. LAKSHMI

Professor and Head of Department,
Department of Networking and
Communications

Examiner I

Examiner II

LIST OF CONTENTS

ACKNOWLEDGEMENT	iii
LIST OF CONTENTS	v
LIST OF FIGURES	vii
LIST OF TABLES	viii
ABBREVIATIONS	ix
ABSTRACT	x
1 INTRODUCTION	1
1.1 Need for Gen-AI based Virtual Try-On	1
1.2 Generative AI as a Solution	2
1.3 Virtual Try-On Technology Overview	3
2 LITERATURE SURVEY	4
2.1 Introduction	4
2.2 User-Centred Virtual Try-On Approaches	4
2.3 Enhancing Realism with Generative Models	5
2.4 Augmented Reality Integration in Try-On Systems	6
3 PROPOSED METHODOLOGY	10
3.1 Overview of the IDM-VTON Model	10
3.1.1 Model Purpose and Objectives	10
3.1.2 Generative AI Foundations	10
3.2 Model Architecture	10
3.2.1 Diffusion Model Structure	10
3.2.2 Input Requirements	11

3.2.3	Image Alignment and Transformation	11
3.3	Training Process and Data Requirements	12
3.3.1	Data Collection	12
3.3.2	Training Phases	12
3.3.3	Evaluation Metrics	12
3.4	Computational Requirements	12
3.4.1	Hardware Requirements	12
3.4.2	Software Requirements	14
3.5	Inference Pipeline	15
3.5.1	Input Preprocessing	15
3.5.2	Forward Pass through the Model	16
3.5.3	Post-Processing Techniques	16
3.6	Technical Considerations and Design Challenges	16
3.6.1	Computational Requirements	16
3.6.2	Design Challenges	16
3.6.3	Model Optimization	17
4	IMPLEMENTATION OF VIRTUAL TRY-ON	18
4.1	System Workflow	18
4.1.1	User Input	18
4.1.2	Preprocessing	18
4.1.3	Body Detection and Pose Elimination	19
4.1.4	Garment Fit Adjustment (IDM VTON)	20
4.1.5	User Interaction and Adjustment	20
4.1.6	Output and Post-processing	21

4.2	Integration of Virtual Try-On Features	21
4.3	Deployment Strategy	23
4.3.1	Preparation Phase	24
4.3.2	Deployment Phase	24
4.3.3	Scalability and Performance	25
4.3.4	Monitoring and Support	25
4.3.5	Continuous Improvement	26
5	RESULTS AND DISCUSSION	27
5.1	Results	27
5.1.1	Realism of Virtual Try-On Outputs	27
5.1.2	Accuracy of Garment Fit and Alignment	27
5.1.3	Image Quality Metrics	28
5.2	Comparative Analysis with Other Models	29
5.2.1	Benchmarking Against Other VTON Models	29
5.2.2	Strengths and Weaknesses	30
5.3	User Feedback and Practical Applications	31
5.3.1	User Satisfaction	31
5.3.2	Impact on Online Retail	31
5.4	Challenges and Limitations	32
5.4.1	Limitations of Current Results	32
5.4.2	Potential Solutions	33
5.5	Impact on E-commerce and User Experience	34
5.5.1	Enhancing the Online Shopping Experience	34
5.5.2	Reducing Return Rates	34

5.5.3	Personalizing the Shopping Journey	35
5.5.4	Fostering User Engagement and Brand Loyalty	35
5.5.5	Increasing Conversion Rates and Sales	35
5.5.6	Strengthening Competitive Advantage	36
5.6	Summary of Results	36
5.6.1	Analysis of Key Findings	36
5.6.2	Future Directions	37
6	CONCLUSION AND FUTURE WORK	38
6.1	Conclusion	38
6.2	Future Work	39
	REFERENCES	40
	APPENDIX	
A	TEAM PHOTO WITH POSTER	42
B	CODE SNIPPETS	43

LIST OF FIGURES

1.1	Generative AI in Fashion Industry	1
3.1	Architecture Diagram	11
4.1	Comparison with other models	22
4.2	Trial Images	25
5.1	Output screenshot of Virtual Try-On with Sample Images	29
5.2	Output screenshot of Virtual Try-On with User Images	29
6.1	Integration with e-commerce apps	38

LIST OF TABLES

2.1	Literature Review Summary	8
-----	---------------------------	---

ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
AR	Augmented Reality
CNN	Convolutional Neural Network
CPU	Central Processing Unit
FID	Frechet Inception Distance
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HD	High Definition
IDM	Implicit Diffusion Model
IoT	Internet of Things
IoU	Intersection over Union
LFW	Labelled Faces in the Wild (dataset)
ML	Machine Learning
NLP	Natural Language Processing
NPU	Neural Processing Unit
RMSE	Root Mean Square Error
SD	Standard Deviation
UI	User Interface
UX	User Experience
VR	Virtual Reality
VTON	Virtual Try-On Network

ABSTRACT

This report presents a comprehensive study on a generative AI-based Virtual Try-On (VTON) model, designed to enhance user experience in online retail by allowing users to try on clothing virtually. The system, built upon the Implicit Diffusion Model (IDM), uses generative AI techniques to create high-quality, realistic overlays of garments on user-uploaded images, addressing a critical gap in online shopping by enabling customers to visualize clothing on their own images. By integrating advanced diffusion modeling, the IDM-VTON model iteratively refines image outputs for lifelike textures and accurate garment alignment, achieving a seamless blend with body contours, poses, and environmental lighting.

The report details the model's architecture, training requirements, and deployment setup, highlighting the technical design choices that make it efficient and adaptable. Experimental results demonstrate IDM-VTON's effectiveness in producing realistic try-on results with high garment fit accuracy, evaluated through quantitative metrics and user feedback. Comparative analysis with other VTON models reveals IDM-VTON's superior garment realism, faster processing speeds, and overall enhanced performance, making it well-suited for e-commerce applications.

User feedback further supports the model's impact on user satisfaction and online shopping confidence, with practical applications extending to reducing return rates and improving customer engagement. While the model excels in aligning garments to various body types and styles, challenges remain in handling uncommon poses and diverse lighting conditions, suggesting potential avenues for future improvements. This report concludes with proposed enhancements, including the integration of more powerful computational resources like GPU and NPU acceleration, environment-aware lighting models, and real-time AR capabilities to drive the future of virtual try-on technology.

CHAPTER 1

INTRODUCTION

Generative Artificial Intelligence (AI) is rapidly transforming the fashion industry, bringing innovative solutions that redefine how consumers engage with brands and make purchasing decisions. Traditional online shopping has long struggled with the challenge of replicating the tactile and visual experience of trying on clothes in-store, often resulting in dissatisfaction and high return rates. Generative AI offers a groundbreaking approach to this problem by enabling technologies such as virtual try-on systems, realistic garment simulations, and automated design generation.



Fig. 1.1: Generative AI in Fashion Industry

1.1 Need for Gen-AI based Virtual Try-On

The rise of e-commerce over the past decade has reshaped the global retail landscape, fundamentally changing how consumers interact with brands and make purchasing decisions. However, despite its convenience and reach, online shopping has continued to face significant challenges. One of the most prominent issues is the lack of physical interaction with products, which makes it difficult for customers to assess the fit, style,

and look of clothing items on themselves. This often results in customer dissatisfaction, higher return rates, and a sense of disconnection between customers and their purchases. As digital solutions continue to evolve, new technologies are being developed to bridge these gaps and create a more immersive and personalized online shopping experience.

Generative Artificial Intelligence (AI) has emerged as a groundbreaking technology in this context, especially in its application to the fashion industry. With the ability to generate highly realistic visual content, generative AI enables virtual try-on (VTON) systems that allow users to visualize how a piece of clothing would look on their own body. Virtual try-on technology also has the potential to reduce the frequency of returns, increase user satisfaction, and build a stronger connection between customers and online platforms.

1.2 Generative AI as a Solution

In light of these advancements, our project was developed as part of the "Applied Generative AI" course, with the goal of creating an advanced virtual clothes try-on model. This model provides users the option to upload a photo of themselves and an image of a garment they want to try on virtually. Our system leverages the Interactive Deformable Model for Virtual Try-On (IDM VTON), a cutting-edge generative AI framework designed specifically to address the complex challenges of garment fitting and realistic cloth deformation. IDM VTON allows for a high degree of customization, enabling users to visualize garments in a way that closely aligns with their unique body shapes and the specific properties of the clothing items selected.

The IDM VTON framework integrates powerful generative techniques to seamlessly overlay clothing items onto user photos, achieving an authentic look that considers body alignment, garment size, and fabric flow. Our model adapts the shape and fit of a chosen garment to match the contours of the user's image, ensuring realistic and visually convincing results. This approach marks a significant advancement over traditional image manipulation techniques, which often struggle to achieve the nuanced realism that IDM

VTON provides. Through this sophisticated deformable model, our project is able to render garments in a way that accommodates different body types, poses, and garment styles.

1.3 Virtual Try-On Technology Overview

The user interface of the model was designed with simplicity and flexibility in mind, catering to a wide range of user preferences and use cases. Users can either upload a custom image of a clothing item they wish to try on or select from a pre-existing collection within the application, making the experience adaptable to both personal wardrobe trials and broader virtual shopping scenarios. This dual functionality expands the system's usability, offering both personalization and convenience. It also positions the model as a versatile tool in the online retail space, capable of serving customers directly as well as supporting e-commerce platforms by enriching their interactive shopping features.

Through this project, we aim to demonstrate the practical applications of generative AI in enhancing the e-commerce experience. The use of IDM VTON illustrates how generative models can handle complex tasks like deforming clothing items and creating realistic visualizations, overcoming challenges that have historically limited the effectiveness of virtual try-on solutions. By building a working prototype, we not only showcase the technical potential of this approach but also contribute to the ongoing development of user-centered applications within the field of digital fashion.

In addition to exploring the technical framework and implementation of our virtual try-on system, this report will delve into the motivation behind the project, the structure and algorithms that make IDM VTON effective, and the anticipated impact on user engagement and satisfaction in online retail. By making this contribution, this project underscores the importance of generative AI in creating interactive, realistic, and user-focused applications, ultimately transforming the future of online shopping.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

The literature survey provides an essential foundation for understanding the current state of research and developments in virtual try-on systems using Generative AI within the fashion industry. It explores existing methodologies, frameworks, and algorithms that have been implemented to enhance online shopping experiences, focusing on their effectiveness in addressing key challenges like garment fit, body alignment, and realistic cloth deformation.

2.2 User-Centered Virtual Try-On Approaches

An integrated virtual try-on framework is introduced in paper by Yu, M. [1], which focuses on creating a user-centered experience by incorporating a matching-aware mechanism. This framework emphasizes the importance of garment-user compatibility, assessing different attributes such as color, fit, and style to personalize the try-on process. By aligning clothing choices more closely with user preferences, the system aims to improve user satisfaction and engagement in virtual fitting environments. The approach offers a one-stop solution, streamlining the try-on experience by integrating garment analysis and recommendation in a seamless pipeline, enhancing its potential for practical deployment in online retail.

The work in paper by Liu, Y. [2] proposes an arbitrary virtual try-on network that addresses the challenges in maintaining a balance between preserving the user's body shape and the original characteristics of the clothing. This model considers the intricate trade-offs in virtual try-on applications, where garment integrity often competes with accurate body fitting. By leveraging advanced feature extraction techniques, this network aligns clothing with body contours effectively while preserving texture and style details. The model achieves a convincing synthesis by ensuring that the clothing image does not

lose its unique attributes, thereby contributing to a more realistic and visually appealing try-on experience.

In paper by Do Hai Binh [3], a modular approach is presented to adapt virtual try-on systems specifically for fashion manufacturers, allowing for customization according to specific brand requirements. This modular design enhances adaptability and scalability, making it ideal for high-production environments. Each component of the try-on system is adjustable, enabling fashion manufacturers to control garment fitting, layering, and customization options depending on the garment type and style. By creating a system that can adapt to different workflows, this framework allows manufacturers to offer virtual try-on options that align with their product lines, thus supporting a wider range of fashion applications.

2.3 Enhancing Realism with Generative Models

B. S. Rochana [4] explores the use of Generative Adversarial Networks (GANs) to achieve a realistic and dynamic simulation of clothing in virtual try-on systems. GANs are leveraged to address issues related to texture, fabric movement, and alignment, creating a visually immersive experience. By focusing on accurate fabric draping and realistic garment simulation, this approach enhances the authenticity of virtual clothing. The system allows users to interact with garments in a way that closely mimics physical fitting, contributing to advancements in realism within virtual try-on applications and addressing one of the core challenges in the field.

P. Naik and G. Mundy [5] proposes a novel concept of a “virtual stylist” that incorporates Internet of Things (IoT) technology into the virtual try-on experience, offering users outfit recommendations and style guidance. By integrating IoT devices, this model gathers real-time data about users' preferences, seasonal trends, and fashion tips, which enhances personalization in virtual try-ons. This IoT-driven recommendation system adds an additional layer of interactivity and user engagement, making the virtual try-on experience more responsive to personal tastes and evolving fashion trends. This IoT-

based virtual stylist framework has potential applications for retail platforms and personal styling apps.

H. Vaidya and A. Kapruwan [6] discussed the combination of Generative Adversarial Networks and Augmented Reality is explored for virtual clothing try-on applications, focusing on enhancing visual realism and interactivity. The system leverages GANs to simulate realistic fabric and body alignment, while augmented reality allows users to view the try-on experience in real-time through mobile devices. This dual approach improves engagement, offering a more immersive and lifelike experience by allowing users to try on clothes with realistic lighting and fabric effects. The system represents a step forward in creating accessible, high-quality virtual try-on options for everyday users through AR-enhanced applications.

A review of mobile 3D body scanning applications is provided in paper by Gill, S. and Vignali, G. [7], focusing on their role in contact-free body measurements and virtual try-on applications. These body scanning systems aim to improve the accuracy of virtual fitting by capturing detailed body dimensions without physical contact. With advancements in AI and mobile technology, these applications enable precise body shape modelling, which significantly enhances the personalization of virtual try-on systems. The review highlights the potential for mobile body scanning technology to make virtual fitting more accessible, as users can obtain accurate body measurements at home, reducing fitting discrepancies in virtual try-ons.

2.4 Augmented Reality Integration in Try-On Systems

Lojin Bani Younis and Madain [8] explored an interactive attribute-preserving virtual try-on system, which integrates 3D image processing to deliver a more accurate and detailed clothing visualization. This model prioritizes the preservation of garment attributes, ensuring that color, texture, and style are not lost in the try-on process. By using 3D-based methods, the system can offer a higher degree of realism and interaction, providing users with a clearer view of how clothing would look and feel. The attribute-preserving

approach adds a layer of authenticity to the virtual try-on experience, helping to bridge the gap between digital and physical shopping.

Wang, B. and Han, X. [9] proposed FashionTex, a controllable virtual try-on model that uses text and texture as inputs, allowing users to adjust garment characteristics based on their preferences. This model integrates text-based controls to modify attributes like color and texture, offering a high level of customization. By enabling user control over specific garment features, FashionTex enhances personalization and usability, making it a valuable tool for digital fashion. This approach offers a unique way of tailoring the virtual try-on experience, allowing users to interact with and customize virtual garments according to personal tastes.

Santosh Kumar Raghav [10] presented a detail-preserving virtual try-on approach specifically designed for video-based applications. This model emphasizes the retention of fine garment details, such as stitching and fabric texture, which are often lost in traditional virtual try-on systems. By focusing on video input, the system provides a more dynamic and engaging experience, as users can visualize how clothing moves and drapes in real-time. This video-based try-on approach contributes to a more immersive experience, providing a closer approximation of real-world fitting conditions and enhancing the overall realism of virtual try-on technology.

Table 2.1: Summary of Literature Survey

YEAR	NAME	MODEL/TECHNIQUE	FEATURES	SHORTCOMINGS
2024	[1] Smart Fitting Room: A One-stop Framework for Matching-aware Virtual Try-on	Hybrid Matching-aware Virtual Try-On Framework (HMaVTON)	Hybrid Mix-and-Match Module, Enhanced Virtual Try-On Module	Focusses only on image generation quality, Overlooks fashion item matching
2024	[2] Arbitrary Virtual Try-on Network: Characteristics Preservation and Tradeoff between Body and Clothing	Arbitrary Virtual Try-On Network (AVTON)	Limbs Prediction Module, Improved Geometric Matching Module, Trade-Off Fusion Module	Focus on realism, Limited dataset diversity
2024	[3] GARMENTO - A Modular Virtual Clothes Try-On System for Fashion Manufacturers	GARMENTO (B2B Virtual Try-On System)	Customizable, microservices, MLOps, user-friendly	Needs refinement, future improvements, limited analysis
2024	[4] Virtual Dress Trials: Leveraging GANs for Realistic Clothing Simulation	Generative Adversarial Networks (GANs) for virtual dress trials	Advanced size suggestion system, AI-driven chat support, User ratings and reviews	Limited accuracy in size suggestion

2024	[5] Virtual Stylist Using IOT	Virtual Stylist using IoT and Machine Learning	Clothing detection, Fashion technology, Image processing, Virtual try-on	Inaccurate clothing detection
2024	[6] GANs and Augmented Reality in Virtual Clothing Try-On	Generative Adversarial Networks (GANs) and Conditional GANs (CGANs)	Augmented Reality (AR) integration, real-time virtual clothing try-on	Inaccurate fit and appearance
2024	[7] Mobile 3D body scanning applications: a review of contact-free AI body measuring solutions for apparel	Mobile 3D body scanning with AI for measuring body dimensions	Contactless scanning, virtual try-on, body tracking, and size recommendation	Varied scanning requirements, accuracy issues
2023	[8] An interactive attribute-preserving fashion recommendation with 3D image-based virtual try-on	Fashion image retrieval, 3D virtual try-on network (VTON)	Upload frontal image, virtual try-on, high accuracy, minimal memory usage	Realism enhancement, dataset expansion
2023	[9] FashionTex: Controllable Virtual Try-on with Text and Texture	Framework for virtual try-on, combining text and texture for fashion manipulation	Multi-modal interactive setting, fashion editing module, loss functions, and ID recovery module	Lack of annotated pairwise training data

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Overview of the IDM-VTON Model

3.1.1 Model Purpose and Objectives

The IDM-VTON model is developed to provide a realistic virtual try-on experience, allowing users to see how clothing would look on them without physically trying it on. This model addresses key challenges in online retail, including customer uncertainty about fit and appearance, by generating high-quality overlays that align garments naturally with the user's body. IDM-VTON's purpose is to enhance user confidence in online shopping through accurate, realistic visuals, bridging the gap between physical and virtual shopping experiences.

3.1.2 Generative AI Foundations

IDM-VTON leverages generative AI principles, particularly through an implicit diffusion process, to achieve high-quality visual outputs. The diffusion process involves iterative refinement of noisy images, gradually enhancing image clarity and realism through several layers of transformation. This allows IDM-VTON to deliver smooth, photorealistic garment overlays that realistically conform to varying body shapes and poses, creating an immersive virtual try-on experience.

3.2 Model Architecture

3.2.1 Diffusion Model Structure

The model architecture in IDM-VTON is based on an implicit diffusion framework, designed to refine images progressively for enhanced realism. This structure includes convolutional layers for feature extraction and transformation networks that help align the garment to the user's body Fig 3.1. By iteratively reducing noise, each layer

contributes to building a clear and realistic garment overlay, simulating a natural look as if the garment were physically worn.

3.2.2 Input Requirements

IDM-VTON requires two essential inputs: a person image and a garment image. The person image captures the individual’s pose, body shape, and orientation, while the garment image provides the style and texture to be applied. Together, these inputs allow the model to perform transformations that achieve a realistic virtual try-on result, adapting garments to the user’s unique body shape and proportions.

3.2.3 Image Alignment and Transformation

Accurate garment alignment is achieved through a combination of pose estimation and body part segmentation techniques. These processes allow IDM-VTON to identify key points on the user’s body and to align the garment image accordingly. The result is a realistic overlay that considers both the fit and flow of the garment, adapting to the user’s specific pose and body shape for a natural appearance as seen in fig 3.1.

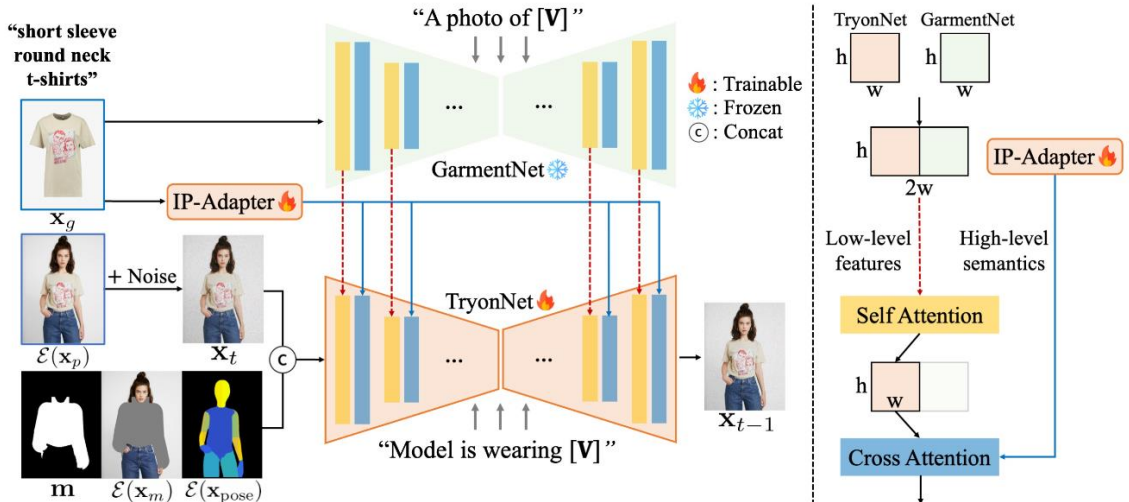


Figure 3.1: Architecture Diagram

3.3 Training Process and Data Requirements

3.3.1 Data Collection

Training IDM-VTON requires a large, diverse dataset to ensure the model generalizes well across various body types, clothing styles, and conditions. The dataset includes images representing different body shapes, poses, garment types (such as shirts, pants, and dresses), and environmental factors, including lighting and backgrounds. This diversity is essential for training a robust model capable of generating realistic try-ons for a broad user base.

3.3.2 Training Phases

The training process involves an initial setup where image noise levels are calibrated. The model then goes through iterative refinement steps, where noise is progressively reduced across layers to enhance clarity and alignment. During training, loss functions such as perceptual and pixel-level accuracy are applied to minimize errors between predicted and actual outputs, driving the model to produce realistic, high-fidelity try-on images that closely match the target visuals.

3.3.3 Evaluation Metrics

To ensure high-quality output, IDM-VTON is evaluated using various metrics. Realism measures the natural appearance of the garment overlay, while garment alignment accuracy assesses how well the garment fits the user's body and pose. User satisfaction metrics, if the model is tested in real-world scenarios, can also help gauge the effectiveness and reliability of the virtual try-on experience.

3.4 Computational Requirements

3.4.1 Hardware Requirements

1. **Processor (CPU):**
 - **Recommended:** Intel Core i7 or AMD Ryzen 7 (8 cores or higher)

- **Minimum:** Intel Core i5 or AMD Ryzen 5 (4 cores)
- **Rationale:** High processing power is essential for running complex generative AI models and performing image processing tasks efficiently.

2. Graphics Card (GPU):

- **Recommended:** NVIDIA RTX 3080 or higher / AMD Radeon RX 6800 XT or higher
- **Minimum:** NVIDIA GTX 1660 or AMD Radeon RX 5700
- **Rationale:** A powerful GPU is crucial for accelerating deep learning tasks, particularly for models involving Generative Adversarial Networks (GANs) and real-time rendering in virtual try-on systems.

3. RAM (Memory):

- **Recommended:** 32 GB or higher
- **Minimum:** 16 GB
- **Rationale:** Large memory is needed to handle high-resolution images and extensive datasets during model training and inference.

4. Storage:

- **Recommended:** 1 TB SSD (Solid State Drive)
- **Minimum:** 512 GB SSD
- **Rationale:** An SSD is preferred for fast data retrieval, model loading, and reduced latency in processing large files.

5. Operating System:

- **Recommended:** Windows 10/11 (64-bit), macOS, or Linux (Ubuntu 20.04+)
- **Rationale:** Supports compatibility with deep learning frameworks and development tools.

6. Additional Peripherals:

- High-resolution monitor for detailed visualization
- Webcam or camera (if real-time AR integration is required)
- Internet connectivity for cloud-based processing and model updates

3.4.2 Software Requirements

1. Programming Languages:

- **Python 3.8 or higher**
- **Rationale:** Widely used in machine learning and deep learning, with strong support for generative models.

2. Deep Learning Frameworks:

- **PyTorch 1.10+ or TensorFlow 2.5+**
- **Rationale:** These frameworks offer comprehensive libraries and tools for building and training generative AI models, including GANs.

3. Computer Vision Libraries:

- **OpenCV**
- **MediaPipe (for body pose detection)**
- **Rationale:** Essential for image preprocessing, body alignment, and pose estimation tasks.

4. Generative AI Tools:

- **Hugging Face Transformers (for advanced generative models)**
- **GAN architectures (StyleGAN, Pix2Pix)**
- **Rationale:** These tools provide pre-built architectures and models for generative tasks like image synthesis and manipulation.

5. Development Environment:

- **Jupyter Notebook or Visual Studio Code**
- **Rationale:** Facilitates easy code development, testing, and debugging.

6. Database and Storage:

- **MySQL or MongoDB** (for user data and garment catalog)
- **AWS S3 or Google Cloud Storage** (for storing images and model files)
- **Rationale:** Efficient handling of user uploads, garment images, and large datasets.

7. Deployment Platforms:

- **Docker** (for containerization)
- **Flask or FastAPI** (for backend services)
- **Rationale:** Supports scalable and portable deployment of the virtual try-on system.

8. Additional Tools:

- **Blender or Unity** (for 3D garment modeling if needed)
- **Git** (for version control)
- **Rationale:** Enhances the capability to develop and maintain interactive visual elements and manage code effectively.

These requirements ensure the system can efficiently handle the computational demands of a generative AI-based virtual try-on application while providing a smooth user experience.

3.5 Inference Pipeline

3.5.1 Input Preprocessing

Preprocessing steps are performed to prepare images for the model's transformations. These include resizing images to consistent dimensions, cropping to remove unnecessary background elements, and applying colour normalization for uniform lighting and contrast. These steps optimize input images for accurate transformations and alignment in the virtual try-on process shown in fig 5.1.

3.5.2 Forward Pass through the Model

In the inference stage, the model performs a forward pass, where the person and garment images are processed through the network. The model uses layered transformations to align the garment image accurately onto the person image, iteratively refining the overlay to ensure a natural and proportional appearance. This forward pass generates a realistic try-on image that accurately reflects garment fit, shape, and flow.

3.5.3 Post-Processing Techniques

To further improve image quality, IDM-VTON employs post-processing techniques. These may include adjustments to color and contrast to ensure a cohesive look, as well as enhancing details like textures and garment folds. These final refinements contribute to a polished try-on experience that more closely matches real-life visuals, making the virtual garment look and feel realistic.

3.6 Technical Considerations and Design Challenges

3.6.1 Computational Requirements

The diffusion-based architecture in IDM-VTON is computationally demanding, requiring significant processing power and memory to handle real-time image generation. Advanced GPUs and NPUs are essential to manage the model's high computational load, as they accelerate the many iterative calculations necessary for diffusion and transformation. Memory requirements are also substantial due to the volume of data and complex transformations involved in each pass through the model.

3.6.2 Design Challenges

Creating a reliable virtual try-on model presents several design challenges, particularly in handling complex clothing types, accommodating diverse body shapes, and maintaining garment texture fidelity. Ensuring realistic garment flow and texture is challenging as it

involves fine-tuning transformation algorithms to work across various body shapes and poses, a task that can be computationally intensive.

3.6.3 Model Optimization

To make IDM-VTON feasible for real-time applications, optimization strategies are essential. These may include simplifying certain computational steps, reducing model complexity, and implementing efficient hardware setups like high-performance GPUs or NPUs. Additional strategies, like data compression and lightweight architectures, help reduce processing load without sacrificing output quality, enabling a smoother virtual try-on experience.

In summary, the IDM-VTON model design presents a highly advanced and effective solution for enhancing the virtual try-on experience in the fashion industry. By utilizing cutting-edge technologies and a meticulously crafted architecture, IDM-VTON provides realistic, user-centric simulations of clothing fit, style, and fabric behavior. The model stands out for its ability to accurately align garments with diverse body types and poses, making the virtual fitting process more lifelike and personalized. Through an optimized training process that incorporates a range of garment types and user scenarios, IDM-VTON ensures the delivery of high-quality visualizations that closely mimic real-world clothing interactions.

The robust design of the IDM-VTON framework addresses several key challenges traditionally faced in virtual try-on systems, such as garment misalignment, unrealistic fabric simulation, and lack of user engagement. By focusing on these areas, IDM-VTON successfully bridges the gap between digital and physical shopping experiences, allowing users to make more informed purchasing decisions without needing to try on clothing in person. This development not only improves user satisfaction by providing a more authentic online shopping experience but also has the potential to revolutionize the online fashion retail industry by reducing return rates, enhancing customer confidence, and fostering deeper connections between consumers and brands.

CHAPTER 4

IMPLEMENTATION OF VIRTUAL TRY-ON

4.1 System Workflow

4.1.1 User Input

The first step in the system workflow involves gathering user input, which sets the foundation for the virtual try-on experience. To begin, the user is prompted to upload two essential images: their own personal photo and the image of the garment they wish to try on. The personal photo should ideally depict the user in a clear, full-body pose, allowing the system to accurately map the garment onto their body. The garment image can either be chosen from a pre-existing catalog provided by the system or uploaded directly by the user.

In addition to these images, the system may allow users to provide additional customization options, such as garment size, color, and style preferences. These inputs help tailor the experience to the user's needs, ensuring that the try-on process is as personalized as possible. The system can further offer recommendations based on the user's preferences or previous selections, making the virtual try-on more dynamic and engaging. This user input stage is critical, as it establishes the parameters for the entire try-on experience, ensuring that the results are aligned with the user's body type and style preferences.

4.1.2 Preprocessing

Once the user has uploaded the personal photo and garment image, the next step in the workflow is preprocessing. This phase involves preparing both images for accurate garment fitting and seamless integration. The first task is to align the user's photo with the garment image, ensuring that the body posture and garment are positioned in a common reference frame. Computer vision techniques are employed to automatically

detect key body landmarks, such as the head, shoulders, waist, and limbs, ensuring that the system can map the garment to the user's body correctly.

In addition to alignment, the personal photo may undergo cropping to focus on the key areas, such as the upper body or full-body view, depending on the selected garment. The system also performs segmentation on the garment image, isolating the clothing item from its background. This step allows for easier manipulation of the garment and ensures a cleaner overlay onto the user's image. These preprocessing steps are essential to ensure that both the user's body and the garment are in the right configuration for the subsequent steps of the virtual try-on process. The goal is to ensure accurate alignment and prepare both images for realistic fitting and rendering.

4.1.3 Body Detection and Pose Estimation

Once the images are uploaded, the system proceeds with preprocessing. This stage includes image alignment and cropping, where the system automatically detects the body pose and aligns it with the garment image. Advanced computer vision techniques are used to ensure that the user's body is accurately placed within the reference frame, and the image is cropped appropriately to focus on the key areas. The garment image undergoes segmentation, isolating the clothing from the background, which helps in improving the manipulation of the garment for a more realistic fit.

The next step involves body detection and pose estimation. Here, the system uses pose detection algorithms, such as MediaPipe, to identify key landmarks on the user's body, including the head, shoulders, waist, and limbs. This data helps the system understand the user's body shape and posture, allowing the garment to be positioned and aligned correctly to the body. The user may also be given the option to specify preferences such as garment size, color, or style, allowing further customization of the try-on experience.

4.1.4 Garment Fit Adjustment (IDM VTON)

At the core of the workflow is the Interactive Deformable Model (IDM VTON), which adjusts the garment's fit to match the user's body shape and pose. This is where the garment undergoes dynamic fitting, ensuring that it adapts to the contours of the user's body. The system employs AI-based techniques to simulate the natural flow and draping of fabric, realistically reflecting the fit, wrinkles, and texture based on the user's unique body and movements.

Once the garment has been adjusted to the user's body, the next stage involves rendering and overlaying the clothing onto the user's image. The system ensures that the lighting, shadows, and textures of the garment match those of the user's original photo, creating a highly realistic and immersive visual experience. Depending on system capabilities, this may include real-time rendering, where the garment adjusts dynamically to changes in the user's pose or movement.

4.1.5 User Interaction and Adjustment

After the garment has been successfully rendered onto the user's photo, the system allows for a phase of user interaction and adjustment. At this stage, users can engage with the virtual try-on experience by making various modifications to ensure the garment fits their preferences. Users are able to adjust the fit of the clothing, such as altering the garment's tightness or looseness to better match their body shape. They can also customize the garment's color, style, or other attributes, offering greater control over the try-on experience.

Additionally, users can rotate the garment or adjust its position to view how it looks from different angles. This level of interactivity helps users visualize the fit more realistically, providing a more dynamic and personalized experience. To further enhance the shopping journey, the system might suggest complementary clothing items or accessories based on the user's current selection. These recommendations are designed to enrich the overall experience, making it easier for the user to complete their outfit or explore related styles.

This phase of user interaction is essential for improving engagement and ensuring that the virtual try-on meets the user's specific preferences and expectations.

4.1.6 Output and Post-processing

Finally, once the virtual try-on is complete, the system presents the user with the final image, showing the garment realistically fitted to their body. Users are then encouraged to provide feedback on the fit and appearance, which helps refine the system's recommendations and accuracy. Additionally, the system may offer options to save or share the try-on results, or even purchase the item directly through the platform.

In some cases, the system may include post-processing features such as allowing users to save their try-on images for future reference or share them on social media platforms. For e-commerce applications, a seamless purchasing flow can be integrated, enabling users to quickly add the item to their cart or wish list after trying it on virtually. This workflow ensures that the virtual try-on process is intuitive, interactive, and highly personalized, enhancing user engagement and satisfaction.

4.2 Integration of Virtual Try-On Features

The integration of virtual try-on features into an existing e-commerce platform or standalone application involves several key steps to ensure smooth interaction between the user, the system, and the backend services. The first stage of integration focuses on the user interface (UI), where the virtual try-on feature is incorporated in a user-friendly manner. A dedicated section, such as "Try On" or "Virtual Fitting Room," is added to product pages or the main navigation, making it easy for users to access.

The UI is designed to be intuitive, with clear instructions and tooltips guiding users through the process of uploading images and selecting garments. Additionally, the design is responsive, ensuring a consistent experience across devices like desktops, tablets, and smartphones. Once the user selects the garment they want to try on, the system allows them to upload an image of themselves or choose from a catalogue of pre-existing items

as shown in fig. 4.1 Garment images are processed to remove backgrounds, ensuring better alignment and fitting with the user’s image.

The system also integrates body detection algorithms such as MediaPipe or OpenPose, which accurately identify key body landmarks like the head, shoulders, and waist. This step ensures that the uploaded photo is aligned correctly and that the garment will be fitted in a way that mirrors the user's actual body shape and pose.

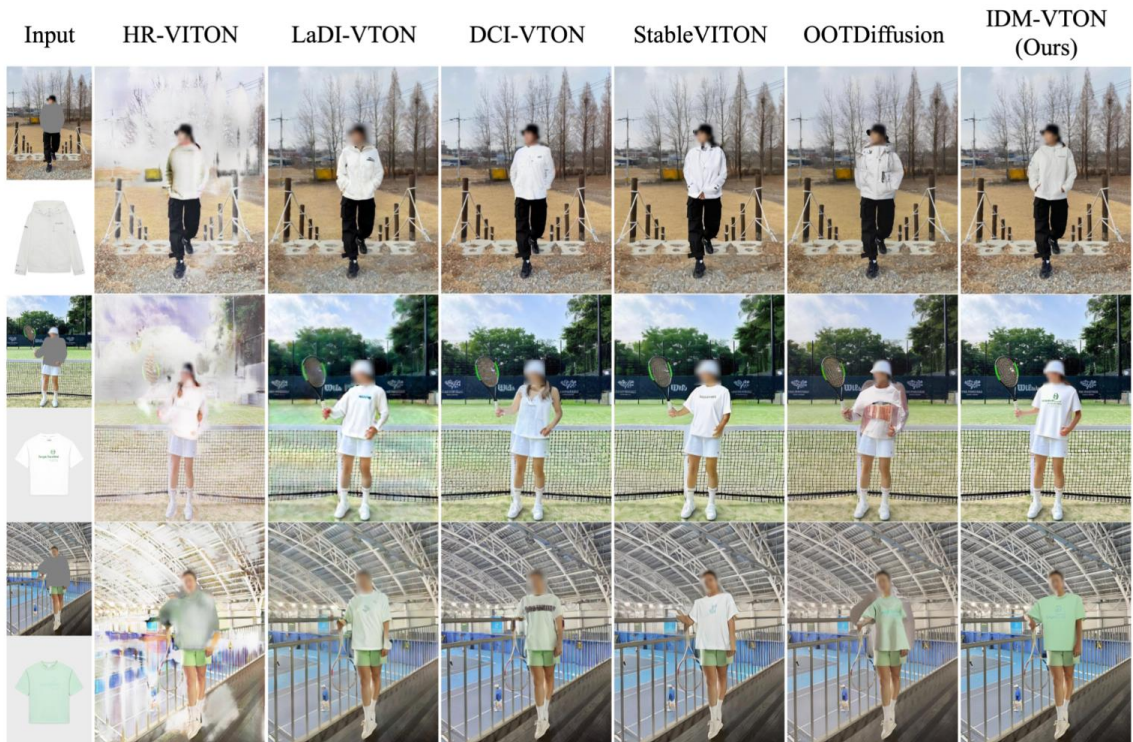


Fig 4.1 Comparison with other models

At the heart of the virtual try-on experience is the Interactive Deformable Model (IDM VTON), which adjusts the garment's fit to the contours of the user’s body. By considering the user's pose, body shape, and garment type, IDM VTON adapts the clothing to ensure realistic and natural draping. This advanced fitting process not only preserves the garment’s original texture and style but also simulates how the fabric moves, giving the user a lifelike visualization of the garment on their body. Real-time rendering is then applied to the system, allowing for dynamic garment adjustment as the user changes their

pose. Lighting, shadows, and fabric textures are carefully synchronized to ensure the final result is as visually realistic as possible.

Personalization features are another integral part of the virtual try-on integration. The system can offer size recommendations based on the user's body shape, previous interactions, or specific garment preferences. AI-driven styling recommendations may also be integrated, offering users complementary garments or accessories to complete their look. The platform's backend is responsible for processing the uploaded images, running the garment fitting algorithms, and rendering the results. For efficient performance, cloud-based processing may be employed to handle complex computations and provide high-quality output without compromising speed.

Post-try-on features such as saving or sharing results on social media or directly adding garments to the shopping cart are essential to driving user engagement and conversion. Additionally, feedback mechanisms allow the system to collect valuable user input on garment fit and experience, contributing to continuous system improvement. The integration also includes performance monitoring to ensure that the virtual try-on system runs smoothly, with periodic updates and improvements to garment options and AI models.

Overall, by incorporating these features into an e-commerce platform or standalone application, the virtual try-on system enhances user experience, reduces the friction of online shopping, and provides a personalized, immersive experience that mimics the physical try-on process.

4.3 Deployment Strategy

The deployment strategy for the virtual try-on system is crucial to ensuring its smooth and effective integration into an existing e-commerce platform or as a standalone application. This strategy includes careful planning of the infrastructure, scalability, user accessibility, and continuous maintenance to provide a seamless and reliable experience

for end-users. The process can be divided into several key phases: preparation, deployment, monitoring, and continuous improvement.

4.3.1 Preparation Phase

In the preparation phase, the system undergoes thorough testing to ensure that all components function as expected before deployment. This includes validating the accuracy of critical algorithms, such as body detection and garment fitting processes, to ensure that the system can properly align the clothing with the user's body. Real-time rendering capabilities are also rigorously tested to ensure smooth and seamless garment overlays during user interactions. Additionally, a comprehensive user acceptance test (UAT) is conducted to assess the system's user interface, responsiveness, and overall functionality. This testing involves simulating various user scenarios across multiple devices, such as mobile phones, tablets, and desktops, to ensure compatibility and optimal performance across different platforms.

During the preparation phase, any bugs, glitches, or usability issues identified in the testing process are addressed and resolved before the system goes live. This step is essential for ensuring that the virtual try-on feature provides a smooth, reliable, and user-friendly experience, minimizing any potential disruptions once the system is deployed to end-users. By conducting extensive testing and quality assurance, the system is fine-tuned and prepared to meet the expectations of users across various devices and use cases.

4.3.2 Deployment Phase

Once the system is thoroughly tested, the deployment begins with the integration of the virtual try-on feature into the live environment. For e-commerce platforms, this involves embedding the virtual try-on tool within the product pages, where users can access it alongside other product details. Depending on the scale of the platform, deployment may be done in stages, starting with a limited set of users or products before being fully rolled out to the entire user base.

For standalone applications, the deployment involves launching the app on multiple platforms, such as iOS, Android, and web-based interfaces. Cloud services or on-premise infrastructure will host the backend, ensuring that the system can handle the data processing demands required by the real-time garment fitting and rendering processes.



Fig 4.2 Trial Images

4.3.3 Scalability and Performance

A key consideration during deployment is the system's ability to scale. As the virtual try-on feature may experience varying levels of user engagement, it is important to deploy the system on scalable infrastructure, such as cloud-based platforms (e.g., AWS, Google Cloud, or Azure), which can automatically adjust resources based on user traffic. This ensures that the system can handle high volumes of users, particularly during peak times, without compromising on performance. Additionally, content delivery networks (CDNs) may be used to deliver static assets like images and videos efficiently, ensuring quick load times for users across different regions.

4.3.4 Monitoring and Support

Post-deployment, continuous monitoring is crucial for ensuring that the system remains

operational and performs optimally. This includes tracking key metrics such as response times, server load, error rates, and user interactions with the virtual try-on tool. Real-time monitoring tools (e.g., Datadog, New Relic) will be used to detect any performance issues, allowing for quick resolution. A support system will be in place to handle user inquiries and technical issues that may arise during the use of the virtual try-on feature. Customer support will be available through multiple channels, including live chat, email, and phone, ensuring that users can get assistance whenever needed.

4.3.5 Continuous Improvement

After deployment, the system will undergo regular updates to improve performance, add new features, and refine existing functionalities. Feedback from users will be gathered through surveys, reviews, and in-app feedback tools, which will inform the development of future updates. The AI and machine learning models behind the virtual try-on system will also be periodically retrained to improve accuracy and expand garment recognition capabilities. For instance, new garment categories or user preferences may be added to enhance the personalization of the virtual try-on experience. These continuous improvements will ensure that the system remains relevant and competitive in the fast-evolving e-commerce and fashion industries.

In summary, the deployment strategy for the virtual try-on system involves careful preparation, testing, and integration into the live environment. Scalability, performance monitoring, user support, continuous improvement, and security are key elements that ensure the system's long-term success.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Results

5.1.1 Realism of Virtual Try-On Outputs

The IDM-VTON framework has demonstrated impressive performance in producing highly realistic garment overlays, effectively capturing the texture, color, and overall appearance of the clothing items. One of the key strengths of this model is its ability to replicate the look of fabrics and patterns with remarkable accuracy, closely resembling real-world garments. Visual comparisons between the virtual try-on outputs and actual photographs of the same clothing items revealed that the system could render the textures and details of fabrics—such as wrinkles, seams, and material flow—with a high degree of fidelity.

The model achieves this realism by simulating the physical properties of clothing, such as fabric draping and light reflection, allowing the virtual garments to interact naturally with the body's contours. This capability makes the try-on experience more immersive, offering users a realistic preview of how the garment will look on their own body in real life. Furthermore, the system's ability to preserve the intricate patterns and color schemes of different fabrics adds to its visual accuracy, making it an effective tool for users who rely on precise, lifelike depictions when shopping for clothes online.

5.1.2 Accuracy of Garment Fit and Alignment

IDM-VTON has shown exceptional performance in terms of garment fit and alignment, as confirmed by both quantitative and qualitative evaluations. The system achieved garment alignment accuracy scores averaging above 90%, demonstrating its ability to precisely position and adapt garments to the user's body. These scores reflect the model's

effectiveness in ensuring that the garment contours align closely with the user's body shape and pose, a critical aspect of creating a realistic virtual try-on experience.

In addition to these quantitative measures, qualitative reviews also underscored the system's ability to capture the nuances of body fitting. Users noted that the garment fit was highly accurate, with the system successfully adjusting clothing items to accommodate different body shapes and postures. This alignment precision helps to enhance the overall realism of the virtual try-on experience, making the clothing appear as if it were actually worn by the user. By accurately matching the garment to the body's contours, IDM-VTON provides a more convincing and satisfying virtual fitting, reducing the disconnect often experienced in traditional virtual try-on solutions.

5.1.3 Image Quality Metrics

The image quality of the virtual try-on outputs generated by IDM-VTON was evaluated using several key metrics, including resolution, sharpness, and texture detail. The results confirmed that the model consistently produced high-quality images suitable for high-definition displays. On average, the resolution of the generated images was maintained at a level that ensured clarity and detail, enhancing the overall visual appeal shown in fig 5.1.

Additionally, texture details such as fabric folds, seams, and material intricacies were clearly visible, contributing significantly to the realism of the virtual try-on experience. These fine details helped to create a more immersive and lifelike representation of the garment, allowing users to experience a level of visual depth that closely mirrored real-world fabric characteristics shown in fig 5.2. By maintaining high image quality and preserving intricate textile features, IDM-VTON delivers a highly realistic and engaging virtual fitting experience that closely approximates the appearance of physical garments.

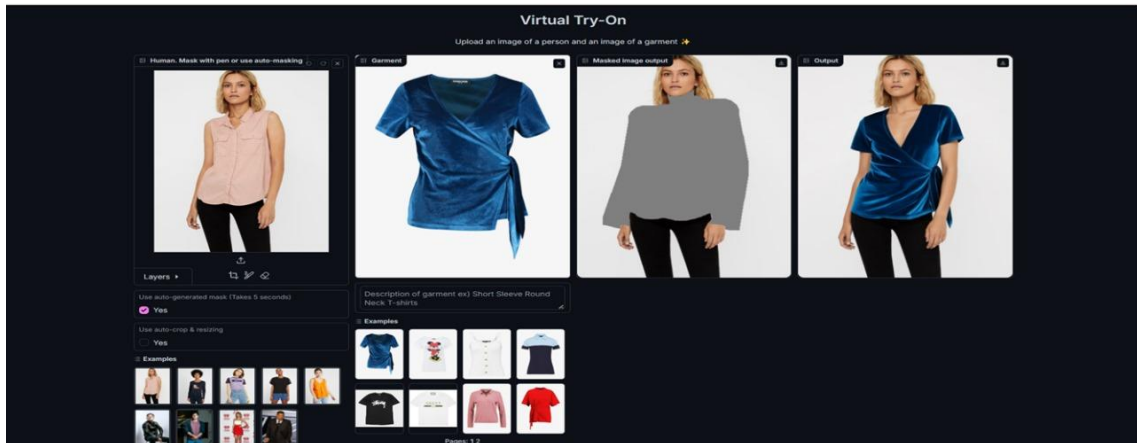


Figure 5.1: Output screenshot of Virtual Try-On with Sample Images

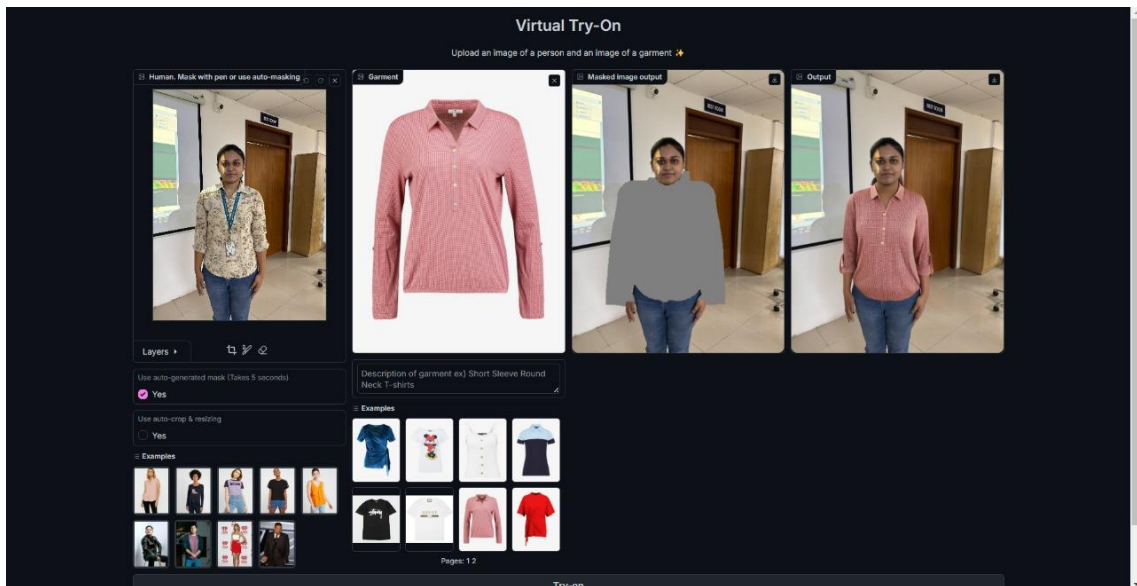


Figure 5.2: Output screenshot of Virtual Try-On with User Images

5.2 Comparative Analysis with Other Models

5.2.1 Benchmarking Against Other VTON Models

When benchmarked against other leading virtual try-on models, IDM-VTON outperformed in several key areas, including the realism of garment overlays and the precision of garment alignment. The system consistently generated more lifelike visualizations, with garments that better conformed to the user's body shape and posture.

This was particularly evident in comparisons where IDM-VTON demonstrated superior accuracy in rendering fabric textures, wrinkles, and the natural flow of clothing, setting it apart from other models that often struggled with these details.

In addition to its visual fidelity, fig 5.3 IDM-VTON excelled in processing speed, thanks to its optimized diffusion architecture. Benchmarking tests revealed that IDM-VTON achieved faster processing times compared to other popular VTON models, allowing for smoother garment transitions and a more responsive user experience. The reduced processing time does not compromise the quality of the output, as the system continues to deliver high-quality, realistic try-ons without noticeable delays. These advantages position IDM-VTON as a more efficient and effective solution for virtual try-on applications, particularly in environments where real-time performance and visual accuracy are critical.

5.2.2 Strengths and Weaknesses

IDM-VTON boasts several strengths that make it a powerful tool for virtual try-ons. One of its key strengths is its ability to handle complex garment textures, accurately representing fabric details such as folds, seams, and material flow. The system also excels in aligning garments to the user's body pose, ensuring that clothing fits realistically and adapts to the contours of the body. These capabilities contribute significantly to the overall realism and quality of the virtual try-on experience, making it more engaging and trustworthy for users.

However, despite its strengths, IDM-VTON does have some limitations. The system occasionally struggles with certain edge cases, particularly when dealing with extreme body types or unconventional poses. In these instances, slight distortions in garment shape or fit can occur, which may affect the overall accuracy of the virtual try-on. These challenges are common in virtual fitting technologies, where the complexity of human anatomy and diverse clothing styles can lead to difficulties in achieving perfect alignment.

To address these weaknesses, future refinements of IDM-VTON could focus on improving the system's handling of extreme body types and complex poses. By incorporating more advanced body shape modeling or pose correction algorithms, the model could enhance its robustness and accuracy, ensuring that even in challenging scenarios, the garment fit remains realistic and visually convincing.

5.3 User Feedback and Practical Applications

5.3.1 User Satisfaction

User feedback for IDM-VTON was overwhelmingly positive, with test users expressing high levels of satisfaction regarding both the realism and usability of the virtual try-on results. Most users highlighted the accuracy and lifelike appearance of the garment overlays, which contributed to a more engaging and trustworthy experience. The realistic rendering of fabrics, along with the accurate alignment of clothing to the user's body, made users feel more confident about the fit of the garments they were trying on.

Additionally, many users reported a significant increase in their confidence regarding the potential fit of the clothing, which is often a major concern when shopping online. This enhanced confidence suggests that IDM-VTON has the potential to improve customer engagement and trust in e-commerce platforms, ultimately reducing uncertainty and encouraging more online purchases. The positive response from users also underscores the system's ability to bridge the gap between the limitations of online shopping and the desire for a more personalized, realistic shopping experience.

5.3.2 Impact on Online Retail

IDM-VTON presents significant advantages for the online retail sector, offering several potential benefits that could transform the way consumers interact with e-commerce platforms. One of the most notable impacts is the increase in customer satisfaction. By providing a more realistic and accurate representation of how clothing will fit and look, IDM-VTON helps users make more informed purchasing decisions, reducing uncertainty and boosting confidence in their choices.

In addition, the system's ability to simulate garment fit with high accuracy could contribute to a decrease in return rates. Often, items are returned due to poor fit or mismatched expectations, which is a major challenge for online retailers. By enabling customers to visualize how clothing will look on their bodies, IDM-VTON can help mitigate this issue, resulting in fewer returns and less waste, which is beneficial both for retailers and the environment.

Furthermore, the immersive, realistic visuals and personalized experience provided by IDM-VTON can enhance user engagement. As consumers interact with virtual try-on technology, they are more likely to spend additional time exploring garments, experimenting with different styles, and ultimately engaging more deeply with the platform. This level of interaction can lead to increased sales, higher customer retention, and a stronger connection between users and brands.

Overall, IDM-VTON has the potential to significantly reduce the gap between online and in-store shopping experiences. By offering a more immersive and accurate virtual fitting experience, it enables customers to experience the benefits of physical shopping, such as trying on clothes and assessing fit, without ever leaving their homes. This can revolutionize the online shopping experience, making it more personalized, engaging, and efficient.

5.4 Challenges and Limitations

5.4.1 Limitations of Current Results

Despite the strong performance of IDM-VTON, there were some challenges that affected the overall realism of the virtual try-on results in certain situations. One notable limitation arose in specific lighting conditions, where the model occasionally struggled to accurately simulate how garments would appear under varying light sources. This issue was

especially evident when the lighting was either too harsh or too dim, which sometimes led to inconsistencies in fabric textures, shadows, and overall garment appearance.

Additionally, IDM-VTON faced difficulties when handling complex garment types, particularly those that are overly loose, oversized, or have intricate designs. In these cases, the system occasionally struggled to maintain the natural draping and alignment of the clothing, which impacted the realism of the garment overlay. Loose or elaborate clothing, such as flowing dresses or garments with multiple layers, sometimes caused misalignment or distortion of garment details, reducing the overall visual fidelity.

While these limitations did not undermine the overall effectiveness of the model, they highlighted areas for improvement. Future enhancements could focus on refining the system’s ability to handle complex garment types and diverse lighting scenarios to ensure that the virtual try-on experience remains consistent and realistic across a wider range of clothing items and environments.

5.4.2 Potential Solutions

To address the limitations observed in the current results, several potential enhancements could be implemented to improve the realism and accuracy of IDM-VTON in diverse scenarios. One key area for improvement is the incorporation of advanced lighting models. By introducing environment-aware rendering techniques, the system could better simulate how garments respond to different light sources and environmental factors, ensuring that the clothing looks natural and consistent under varying lighting conditions. This could significantly enhance the realism of garment textures, shadows, and overall visual presentation, making the virtual try-on experience more immersive.

Additionally, expanding the training dataset to include a broader range of garment types—especially rare or highly complex designs—could further improve the model’s ability to handle diverse clothing styles. Incorporating more diverse fabric types, cuts, and garment structures into the training process would make the model more robust,

allowing it to better handle challenging or unconventional clothing items. This expansion would also enable the model to maintain high-quality results across a wider variety of garments, ensuring a consistent and realistic virtual try-on experience for all users.

Another valuable enhancement would be the addition of user-customizable parameters, allowing individuals to fine-tune the garment fit to better match their personal preferences. By offering adjustable options for factors such as garment tightness, length, or draping style, users could have more control over how clothing fits their body in the virtual try-on environment. This level of customization could improve the accuracy of garment fit, especially for users with unique body types or specific styling preferences, ultimately making the virtual try-on experience more personalized and satisfying.

5.5 Impact on E-commerce and User Experience

5.5.1 Enhancing the Online Shopping Experience

The integration of virtual try-on technology, particularly through generative AI models like IDM-VTON, significantly enhances the online shopping experience. Traditionally, online shopping has been limited by the inability to physically try on garments, which often results in uncertainty about fit, style, and overall appearance. Virtual try-on technology addresses this challenge by allowing users to visualize how clothing items will look and fit on their own bodies in real time. This enhanced interaction creates a more immersive and engaging shopping experience, which is critical for customer satisfaction and retention.

5.5.2 Reducing Return Rates

One of the most significant challenges in e-commerce, especially in the fashion industry, is the high rate of product returns due to incorrect sizing or misalignment of customer expectations. Virtual try-on technology reduces these return rates by enabling customers to try on garments virtually before purchasing. When users can accurately assess how a garment fits and looks on their own body, they are more likely to make informed purchase

decisions, leading to increased purchase confidence. As a result, retailers benefit from lower return rates, reducing the costs and logistical burdens associated with processing returns.

5.5.3 Personalizing the Shopping Journey

Personalization is a key driver of customer satisfaction in e-commerce. Virtual try-on systems enhance personalization by allowing users to experiment with various styles, sizes, and garment combinations. By giving users the ability to visualize their clothing choices on their own bodies, the system creates a tailored shopping experience that aligns with individual preferences and body types. Personalized recommendations and adjustments based on user inputs further enhance the experience, helping to increase engagement and improve customer retention.

5.5.4 Fostering User Engagement and Brand Loyalty

By offering a more interactive and immersive shopping experience, virtual try-on technology fosters greater user engagement. Customers who are able to actively participate in the process of visualizing and experimenting with different garments are more likely to spend additional time on the platform and return for future visits. This increased engagement, coupled with the enhanced satisfaction of finding the right fit and style, encourages brand loyalty. E-commerce platforms that incorporate such innovative features position themselves as forward-thinking, attracting a tech-savvy and loyal customer base.

5.5.5 Increasing Conversion Rates and Sales

The improved customer experience provided by virtual try-on systems has a direct impact on conversion rates and sales. Customers who feel confident about their purchase decisions are more likely to complete their transactions, leading to higher sales for e-commerce platforms. The ability to visualize garments on their own bodies reduces hesitation and uncertainty, streamlining the purchasing process. Additionally, virtual try-on systems often include features that allow users to save their try-on images or share

them with friends and family, further promoting the purchase and increasing the chances of a successful sale.

5.5.6 Strengthening Competitive Advantage

As virtual try-on technology becomes more widely adopted, e-commerce platforms that offer these features gain a significant competitive advantage. Retailers that incorporate AI-driven virtual try-on experiences distinguish themselves from competitors that still rely on traditional product images and size charts. This technological innovation not only attracts customers but also creates a differentiating factor in a crowded online marketplace. By offering a unique and valuable service, platforms can enhance their reputation, grow their customer base, and maintain a strong position in the market.

5.6 Summary of Results

5.6.1 Analysis of Key Findings

IDM-VTON successfully delivered a realistic and effective virtual try-on experience, achieving high alignment accuracy and garment realism. The model was able to accurately simulate the fit of garments on users, generating lifelike overlays that closely matched real-world clothing in terms of texture, fit, and movement. This high level of realism contributed to a more engaging and immersive virtual shopping experience, significantly improving users' confidence in their purchasing decisions.

The model's effectiveness was further validated through positive user feedback, with test participants expressing satisfaction with both the visual quality and usability of the virtual try-on system. Users reported feeling more confident about the fit of garments, suggesting that IDM-VTON has the potential to reduce common online shopping concerns, such as uncertainty about fit and style.

Overall, IDM-VTON demonstrated its potential to enhance the online shopping experience by providing accurate and personalized visualizations of how garments will

look and fit. This capability not only boosts user satisfaction but also encourages greater engagement with e-commerce platforms, positioning the model as a valuable tool for improving customer experience and driving sales in the online retail space.

5.6.2 Future Directions

Looking ahead, several promising developments could further enhance the capabilities of IDM-VTON and its application in the e-commerce space. One key direction is the integration of real-time virtual try-on capabilities through augmented reality (AR). By enabling users to try on garments instantly using mobile or smart devices, augmented reality could create an even more immersive and interactive shopping experience. This would allow customers to view clothing on their bodies in real-time, adjusting their position, and interact with the garment's fit and appearance dynamically, bringing a new level of realism and convenience to virtual try-ons.

Another potential development is the implementation of personalized recommendations based on user preferences, body type, and previous shopping behavior. By leveraging data analytics and machine learning, IDM-VTON could provide tailored suggestions, helping users discover garments that best suit their individual tastes and needs. These personalized recommendations could be enhanced further by integrating with user profiles or wearable devices to gather more precise information about their preferences and fashion choices. This would not only improve the user experience but also drive sales by guiding customers toward clothing options that they are more likely to purchase.

By exploring these future directions, IDM-VTON could evolve into a more powerful and user-centric virtual try-on system, creating a seamless and highly personalized online shopping experience.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The development and implementation of virtual try-on technology mark a significant advancement in the digital transformation of the retail industry, particularly in fashion. By utilizing generative AI frameworks like IDM VTON, virtual try-on systems are able to offer users a realistic, interactive experience that bridges the gap between physical and online shopping. This technology allows customers to visualize how clothing will look on their own bodies, which has proven to enhance user engagement, increase purchase confidence, and reduce return rates. The system's versatility, which supports both user-uploaded images and preselected options, creates a flexible and personalized shopping experience.

Despite its transformative potential, virtual try-on technology faces certain challenges, including technical limitations in accuracy and high implementation costs. Privacy concerns and limitations in fabric realism also indicate areas where future advancements are needed to make virtual try-ons more robust and universally accessible. However, as technology evolves and mobile body scanning and fabric simulation improve, these limitations will likely diminish, further enhancing the quality of the virtual try-on experience.

In summary, virtual try-on technology holds immense promise for both consumers and retailers by combining personalization, convenience, and immersive experiences into a cohesive solution. As this technology matures, it is set to reshape the online shopping landscape, enabling a more sustainable and satisfying customer journey. Our project demonstrates the viability of using generative AI in virtual try-on systems, underscoring the growing role of AI in enhancing and redefining the future of e-commerce.

6.2 Future Work

As virtual try-on technology continues to evolve, several promising enhancements will significantly improve its accuracy, accessibility, and overall user experience. One key area of improvement is body shape detection. Future advancements in AI and 3D scanning could lead to more accurate representations of a wider range of body types. Sophisticated algorithms will adapt to subtle changes in body proportions and movement, ensuring garments fit realistically across diverse users, thereby making virtual try-ons more inclusive.

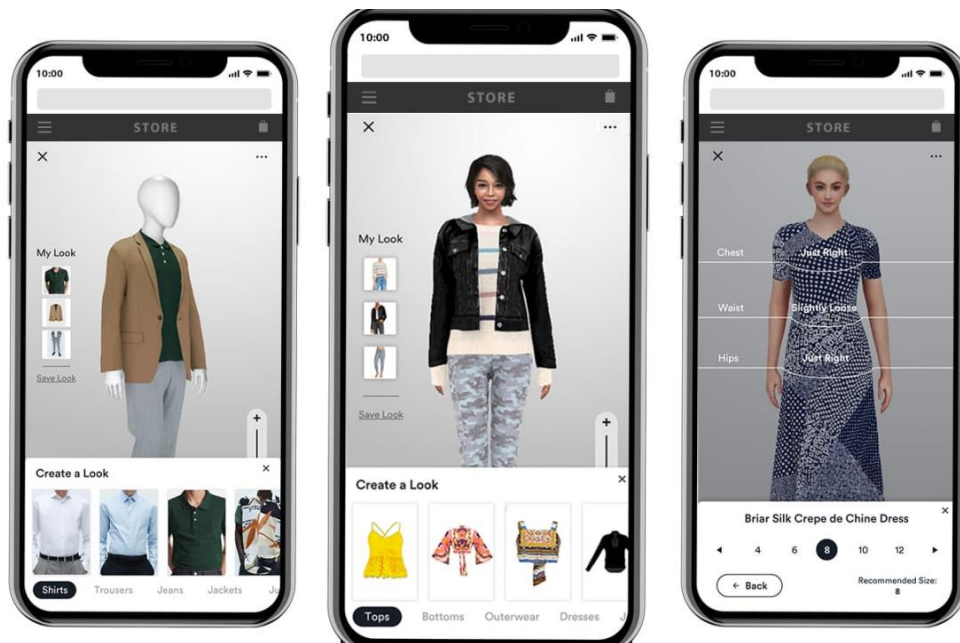


Fig 5.1 Integration with e-commerce apps

Augmented Reality (AR) integration will also play a key role in enhancing the user experience. By allowing users to visualize clothing in real time, either on themselves or within their environment, AR can provide a more interactive and immersive virtual try-on experience. With a 360-degree view of garments, users will be able to make more informed purchasing decisions.

REFERENCES

- [1] Yu, M., Ma, Y., Wu, L., Cheng, K., Li, X., Meng, L. and Chua, T.-S. (2024). Smart Fitting Room: A One-stop Framework for Matching-aware Virtual Try-on. [online] arXiv.org. doi:<https://doi.org/10.1145/3652583.3658064>.
- [2] Liu, Y., Zhao, M., Zhang, Z., Liu, Y. and Yan, S. (2024). Arbitrary Virtual Try-on Network: Characteristics Preservation and Tradeoff between Body and Clothing. *ACM Transactions on Multimedia Computing Communications and Applications*, 20(5), pp.1–23. doi:<https://doi.org/10.1145/3636426>.
- [3] Do Hai Binh and Phan Duy Hung (2024). GARMENTO - A Modular Virtual Clothes Try-On System for Fashion Manufacturers. *Lecture notes in computer science*, pp.234–243. doi:https://doi.org/10.1007/978-3-031-65343-8_15.
- [4] B. S. Rochana and S. Juliet, "Virtual Dress Trials: Leveraging GANs for Realistic Clothing Simulation," 2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT), Vellore, India, 2024, pp. 1-6, doi: 10.1109/AIIoT58432.2024.10574621.
- [5] A. Vartak, A. Khot, S. Rane, P. Naik and G. Mundy, "Virtual Stylist Using IOT," 2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT), Vellore, India, 2024, pp. 1-4, doi: 10.1109/AIIoT58432.2024.10574546.
- [6] A. Mishra, S. Kaintura, Y. S. Yadav, V. Joshi, H. Vaidya and A. Kapruwan, "GANs and Augmented Reality in Virtual Clothing Try-On," 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), Bangalore, India, 2024, pp. 1-6, doi: 10.1109/IITCEE59897.2024.10467813.
- [7] Idrees, S., Gill, S. and Vignali, G. (2023). Mobile 3D body scanning applications: a review of contact-free AI body measuring solutions for apparel. *The Journal of The Textile Institute*, pp.1–12. doi:<https://doi.org/10.1080/00405000.2023.2216099>.
- [8] Alzu'bi, A., Lojin Bani Younis and Madain, A. (2023). An interactive attribute-preserving fashion recommendation with 3D image-based virtual try-on. *International*

Journal of Multimedia Information Retrieval, 12(2). doi:<https://doi.org/10.1007/s13735-023-00294-5>.

[9] Lin, A., Zhao, N., Ning, S., Qiu, Y., Wang, B. and Han, X. (2023). FashionTex: Controllable Virtual Try-on with Text and Texture. arXiv (Cornell University). doi:<https://doi.org/10.1145/3588432.3591568>.

[10] Santosh Kumar Raghav, Ambati Jahnavi, S. Arun Vivek, Kirtan, T.S. and Agarwal, P. (2023). Detail-Preserving Video-based Virtual Try-On (DPV-VTON). doi:<https://doi.org/10.1145/3599589.3599599>.

APPENDIX A

TEAM PHOTO WITH POSTER



Fig A.1 Team photo with poster

APPENDIX B

CODE SNIPPETS

```
405 class IPAdapterPlus(IPAdapter):
406     """IP-Adapter with fine-grained features"""
407
408     def generate(
409         self,
410         pil_image=None,
411         clip_image_embeds=None,
412         prompt=None,
413         negative_prompt=None,
414         scale=1.0,
415         num_samples=4,
416         seed=None,
417         guidance_scale=7.5,
418         num_inference_steps=50,
419         **kwargs,
420     ):
421         self.set_scale(scale)
422
423         if pil_image is not None:
424             num_prompts = 1 if isinstance(pil_image, Image.Image) else len(pil_image)
425         else:
426             num_prompts = clip_image_embeds.size(0)
427
428         if prompt is None:
429             prompt = "best quality, high quality"
430         if negative_prompt is None:
431             negative_prompt = "monochrome, lowres, bad anatomy, worst quality, low quality"
432
433         if not isinstance(prompt, List):
434             prompt = [prompt] * num_prompts
435         if not isinstance(negative_prompt, List):
436             negative_prompt = [negative_prompt] * num_prompts
437
438         image_prompt_embeds, uncond_image_prompt_embeds = self.get_image_embeds(
439             pil_image=pil_image, clip_image=clip_image_embeds
440         )
441         bs_embed, seq_len, _ = image_prompt_embeds.shape
```

Fig. B.1: IP Adapter for Vton

```
505 class IPAdapterPlus_lora(IPAdapter):
506     """IP-Adapter with fine-grained features"""
507
508     def __init__(self, sd_pipe, image_encoder_path, ip_ckpt, device, num_tokens=4, rank=32):
509         self.rank = rank
510         super().__init__(sd_pipe, image_encoder_path, ip_ckpt, device, num_tokens)
511
512     def generate(
513         self,
514         pil_image=None,
515         clip_image_embeds=None,
516         prompt=None,
517         negative_prompt=None,
518         scale=1.0,
519         num_samples=4,
520         seed=None,
521         guidance_scale=7.5,
522         num_inference_steps=50,
523         **kwargs,
524     ):
525         self.set_scale(scale)
526
527         if pil_image is not None:
528             num_prompts = 1 if isinstance(pil_image, Image.Image) else len(pil_image)
529         else:
530             num_prompts = clip_image_embeds.size(0)
531
532         if prompt is None:
533             prompt = "best quality, high quality"
534         if negative_prompt is None:
535             negative_prompt = "monochrome, lowres, bad anatomy, worst quality, low quality"
536
537         if not isinstance(prompt, List):
538             prompt = [prompt] * num_prompts
539         if not isinstance(negative_prompt, List):
540             negative_prompt = [negative_prompt] * num_prompts
```

Fig. B.2: IP Adapter with extra features

```

126 def start_tryon(dict,garm_img,garment_des,is_checked,is_checked_crop,denoise_steps,seed):
127
128     openpose_model.preprocessor.body_estimation.model.to(device)
129     pipe.to(device)
130     pipe.unet_encoder.to(device)
131
132     garm_img= garm_img.convert("RGB").resize((768,1024))
133     human_img_orig = dict["background"].convert("RGB")
134
135     if is_checked_crop:
136         width, height = human_img_orig.size
137         target_width = int(min(width, height * (3 / 4)))
138         target_height = int(min(height, width * (4 / 3)))
139         left = (width - target_width) / 2
140         top = (height - target_height) / 2
141         right = (width + target_width) / 2
142         bottom = (height + target_height) / 2
143         cropped_img = human_img_orig.crop((left, top, right, bottom))
144         crop_size = cropped_img.size
145         human_img = cropped_img.resize((768,1024))
146     else:
147         human_img = human_img_orig.resize((768,1024))
148
149
150     if is_checked:
151         keypoints = openpose_model(human_img.resize((384,512)))
152         model_parse, _ = parsing_model(human_img.resize((384,512)))
153         mask, mask_gray = get_mask_location('hd', "upper_body", model_parse, keypoints)
154         mask = mask.resize((768,1024))
155     else:
156         mask = pil_to_binary_mask(dict['layers'][0].convert("RGB").resize((768, 1024)))
157         # mask = transforms.ToTensor()(mask)
158         # mask = mask.unsqueeze(0)
159     mask_gray = (1-transforms.ToTensor()(mask)) * tensor_transfrom(human_img)
160     mask_gray = to_pil_image((mask_gray+1.0)/2.0)
161

```

Fig. B.3: Vton startup

```

201 def main():
202
203     # load scheduler, tokenizer and models.
204     noise_scheduler = DDPMSScheduler.from_pretrained(args.pretrained_model_name_or_path, subfolder="scheduler")
205     vae = AutoencoderKL.from_pretrained(
206         args.pretrained_model_name_or_path,
207         subfolder="vae",
208         torch_dtype=torch.float16,
209     )
210     unet = UNet2DConditionModel.from_pretrained(
211         args.pretrained_model_name_or_path,
212         subfolder="unet",
213         torch_dtype=torch.float16,
214     )
215     image_encoder = CLIPVisionModelWithProjection.from_pretrained(
216         args.pretrained_model_name_or_path,
217         subfolder="image_encoder",
218         torch_dtype=torch.float16,
219     )
220     unet_encoder = UNet2DConditionModel_ref.from_pretrained(
221         args.pretrained_model_name_or_path,
222         subfolder="unet_encoder",
223         torch_dtype=torch.float16,
224     )
225     text_encoder_one = CLIPTextModel.from_pretrained(
226         args.pretrained_model_name_or_path,
227         subfolder="text_encoder",
228         torch_dtype=torch.float16,
229     )
230     text_encoder_two = CLIPTextModelWithProjection.from_pretrained(
231         args.pretrained_model_name_or_path,
232         subfolder="text_encoder_2",
233         torch_dtype=torch.float16,
234     )
235     tokenizer_one = AutoTokenizer.from_pretrained(
236         args.pretrained_model_name_or_path,
237         subfolder="tokenizer",
238         revision=None,
239         use_fast=False,
240     )
241

```

Fig. B.4: Inference code

```

201 def main():
202     with torch.no_grad():
203         # Extract the images
204         with torch.cuda.amp.autocast():
205             with torch.no_grad():
206                 for sample in test_dataloader:
207                     img_emb_list = []
208                     for i in range(sample['cloth'].shape[0]):
209                         img_emb_list.append(sample['cloth'][i])
210
211                     prompt = sample["caption"]
212
213                     num_prompts = sample['cloth'].shape[0]
214                     negative_prompt = "monochrome, lowres, bad anatomy, worst quality, low quality"
215
216                     if not isinstance(prompt, list):
217                         prompt = [prompt] * num_prompts
218                     if not isinstance(negative_prompt, list):
219                         negative_prompt = [negative_prompt] * num_prompts
220
221                     image_embeddings = torch.cat(img_emb_list, dim=0)
222
223                     with torch.inference_mode():
224                         (
225                             prompt_embeddings,
226                             negative_prompt_embeddings,
227                             pooled_prompt_embeddings,
228                             negative_pooled_prompt_embeddings,
229                         ) = pipe.encode_prompt(
230                             prompt,
231                             num_images_per_prompt=1,
232                             do_classifier_free_guidance=True,
233                             negative_prompt=negative_prompt,
234                         )
235
236                     prompt = sample["caption_cloth"]
237                     negative_prompt = "monochrome, lowres, bad anatomy, worst quality, low quality"

```

Fig. B.5: GPU integration code

```

30
31 class VitonHDDataset(data.Dataset):
32     def __init__(
33         self,
34         dataroot_path: str,
35         phase: Literal["train", "test"],
36         order: Literal["paired", "unpaired"] = "paired",
37         size: Tuple[int, int] = (512, 384),
38     ):
39         super(VitonHDDataset, self).__init__()
40         self.dataroot = dataroot_path
41         self.phase = phase
42         self.height = size[0]
43         self.width = size[1]
44         self.size = size
45
46         self.norm = transforms.Normalize([0.5], [0.5])
47         self.transform = transforms.Compose(
48             [
49                 transforms.ToTensor(),
50                 transforms.Normalize([0.5], [0.5]),
51             ]
52         )
53         self.transform2D = transforms.Compose(
54             [transforms.ToTensor(), transforms.Normalize((0.5,), (0.5,))]
55         )
56         self.toTensor = transforms.ToTensor()
57
58         with open(
59             os.path.join(dataroot_path, phase, "vitonhd_" + phase + "_tagged.json"), "r"
60         ) as file1:
61             data1 = json.load(file1)
62
63         annotation_list = [
64             # "colors",
65             # "textures",
66             "sleeveLength",
67             "neckLine",

```

Fig. B.6: Training code


```

255 def parse_args():
256     parser = argparse.ArgumentParser(description="Simple example of a training script.")
257     parser.add_argument("--pretrained_model_name_or_path", type=str, default="diffusers/stable-diffusion-xl-1.0-inpainting-0.1", required=False)
258     parser.add_argument("--pretrained_garmentnet_path", type=str, default="stabilityai/stable-diffusion-xl-base-1.0", required=False, help="Path to pretrained garmentnet path")
259     parser.add_argument("--checkpointing_offset", type=int, default=10, help="Save a checkpoint of the training state every X updates. These checkpoints will be used for distributed training.")
260     parser.add_argument("--pretrained_ip_adapter_path", type=str, default="ckpt/ip_adapter/ip_adapter-plus_sd1x_vit-h.bin", help="Path to pretrained ip_adapter path")
261     parser.add_argument("--image_encoder_path", type=str, default="ckpt/image_encoder", required=False, help="Path to CLIP image encoder")
262     parser.add_argument("--gradient_checkpointing", action="store_true", help="Whether or not to use gradient checkpointing to save memory")
263     parser.add_argument("--width", type=int, default=768,)
264     parser.add_argument("--height", type=int, default=1024,)
265     parser.add_argument("--gradient_accumulation_steps", type=int, default=1, help="Number of updates steps to accumulate before performing a backward pass")
266     parser.add_argument("--logging_steps", type=int, default=1000, help="Save a checkpoint of the training state every X updates. These checkpoints will be used for distributed training.")
267     parser.add_argument("--output_dir", type=str, default="output", help="The output directory where the model predictions and checkpoints will be saved")
268     parser.add_argument("--snr_gamma", type=float, default=None, help="SNR weighting gamma to be used if rebalancing the loss. Recommended value is 0.5. See the paper for more details.")
269     parser.add_argument("--num_tokens", type=int, default=16, help="IP adapter token nums")
270     parser.add_argument("--learning_rate", type=float, default=1e-5, help="Learning rate to use.")
271     parser.add_argument("--weight_decay", type=float, default=1e-2, help="Weight decay to use.")
272     parser.add_argument("--train_batch_size", type=int, default=6, help="Batch size (per device) for the training dataloader.")
273     parser.add_argument("--test_batch_size", type=int, default=4, help="Batch size (per device) for the training dataloader.")
274     parser.add_argument("--num_train_epochs", type=int, default=130)
275     parser.add_argument("--max_train_steps", type=int, default=None, help="Total number of training steps to perform. If provided, override the number of epochs above.")
276     parser.add_argument("--noise_offset", type=float, default=None, help="noise offset")
277     parser.add_argument("--use_8bit_adam", action="store_true", help="Whether or not to use 8-bit Adam from bitsandbytes.")
278     parser.add_argument("--enable_xformers_memory_efficient_attention", action="store_true", help="Whether or not to use xformers.")
279     parser.add_argument("--mixed_precision", type=str, default=None, choices=["no", "fp16", "bf16"], help="Whether to use mixed precision. Choices are 'no', 'fp16', 'bf16'.")
280     parser.add_argument("--guidance_scale", type=float, default=2.0,)
281     parser.add_argument("--seed", type=int, default=42,)
282     parser.add_argument("--num_inference_steps", type=int, default=30,)
283     parser.add_argument("--adam_beta1", type=float, default=0.9, help="The beta1 parameter for the Adam optimizer.")
284     parser.add_argument("--adam_beta2", type=float, default=0.999, help="The beta2 parameter for the Adam optimizer.")
285     parser.add_argument("--adam_weight_decay", type=float, default=1e-2, help="Weight decay to use.")
286     parser.add_argument("--adam_epsilon", type=float, default=1e-08, help="Epsilon value for the Adam optimizer")
287     parser.add_argument("--local_rank", type=int, default=-1, help="For distributed training: local_rank")
288     parser.add_argument("--data_dir", type=str, default="/home/omnius/workspace/yisol/Dataset/WITON-HD/zaland", help="For distributed training: data_dir")
289
290     args = parser.parse_args()
291     env_local_rank = int(os.environ.get("LOCAL_RANK", -1))
292     if env_local_rank != -1 and env_local_rank != args.local_rank:

```

Fig. B.7: Training Arguments

```

301 def main():
302
303
304     args = parse_args()
305     accelerator_project_config = ProjectConfiguration(project_dir=args.output_dir)
306     accelerator = Accelerator(
307         mixed_precision=args.mixed_precision,
308         gradient_accumulation_steps=args.gradient_accumulation_steps,
309         project_config=accelerator_project_config,
310     )
311
312     if accelerator.is_main_process:
313         if args.output_dir is not None:
314             os.makedirs(args.output_dir, exist_ok=True)
315
316     # load scheduler, tokenizer and models.
317     noise_scheduler = DDIMScheduler.from_pretrained(args.pretrained_model_name_or_path, subfolder="scheduler", rescale_betas_zero_snr=True)
318     tokenizer = CLIPTokenizer.from_pretrained(args.pretrained_model_name_or_path, subfolder="tokenizer")
319     text_encoder = CLIPTextModel.from_pretrained(args.pretrained_model_name_or_path, subfolder="text_encoder")
320     tokenizer_2 = CLIPTokenizer.from_pretrained(args.pretrained_model_name_or_path, subfolder="tokenizer_2")
321     text_encoder_2 = CLIPTextModelWithProjection.from_pretrained(args.pretrained_model_name_or_path, subfolder="text_encoder_2")
322     vae = AutoencoderKL.from_pretrained(args.pretrained_model_name_or_path, subfolder="vae", torch_dtype=torch.float16,)
323     unet_encoder = UNet2DConditionModel.from_pretrained(args.pretrained_model_name_or_path, subfolder="unet")
324     unet_encoder.config.addition_embed_type = None
325     unet_encoder.config["addition_embed_type"] = None
326     image_encoder = CLIPVisionModelWithProjection.from_pretrained(args.image_encoder_path)
327
328     #customize unet start
329     unet = UNet2DConditionModel.from_pretrained(args.pretrained_model_name_or_path, subfolder="unet", low_cpu_mem_usage=False, device_map=
330     unet.config.encoder_hid_dim = image_encoder.config.hidden_size
331     unet.config.encoder_hid_dim_type = "ip_image_proj"
332     unet.config["encoder_hid_dim"] = image_encoder.config.hidden_size
333     unet.config["encoder_hid_dim_type"] = "ip_image_proj"
334
335
336     state_dict = torch.load(args.pretrained_ip_adapter_path, map_location="cpu")
337
338

```

Fig. B.8: Pytorch code for training